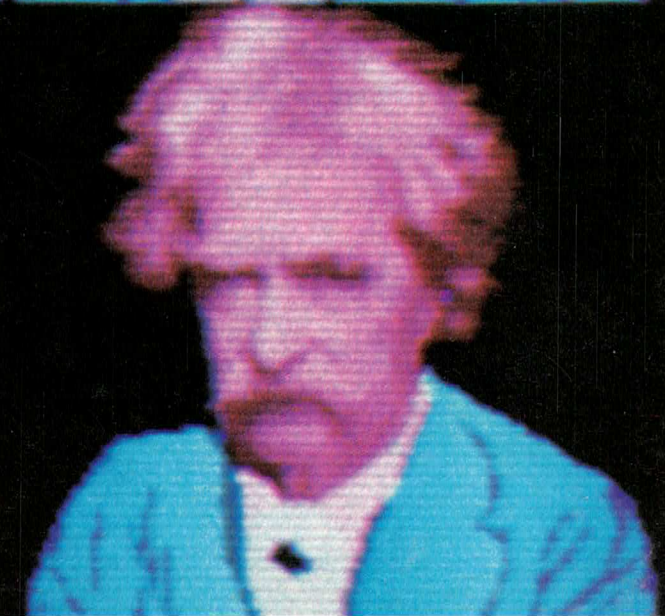
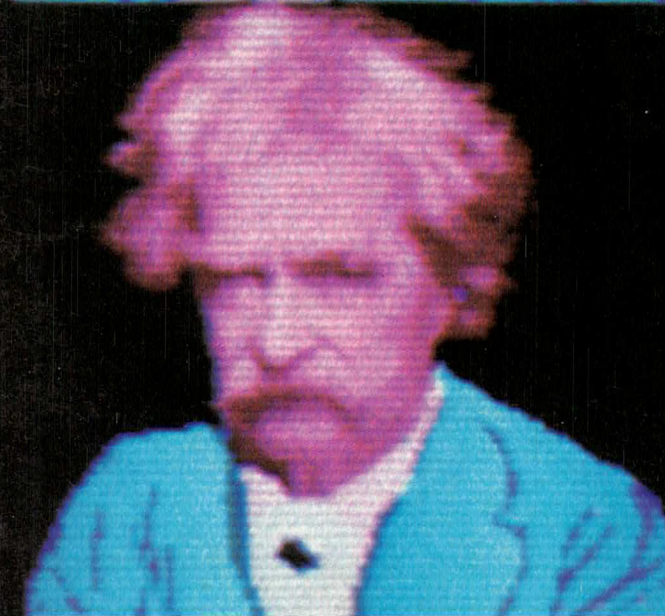
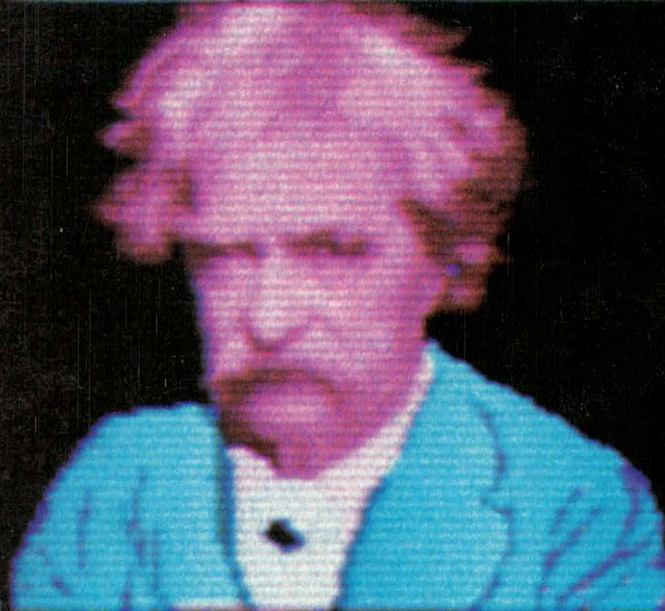


UNIXTM REVIEW

THE PUBLICATION FOR THE UNIXTM COMMUNITY

August 1984

\$3.95



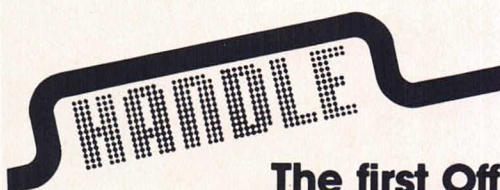
HANDLE[™], the office automation software selected for distribution by AT&T.

Handle is the first office automation or "integrated" software system available for the UNIX[™] operating system. Selected for distribution by **AT&T** Information Systems on the 3B2/300 and 3B5 UNIX systems, **Handle** is the most full-featured office automation system available in the market today. **Handle** offers:

- Dynamically integrated word processing, graphics, list management, virtual spreadsheet, and spelling correction with a powerful "document metaphor" that provides a common method for creating, printing, merging, archiving off-line, filing under multiple topics and retrieving documents.
- Full multi-user capabilities with extensive security — documents or portions of documents can be marked "author only", "read only" or "general access". **Handle** provides its own record locking.
- User preferences — **Handle** conforms to each individual's preferences for such things as preferred printers and individual document catalogs. User restrictions can control such things as the ability to delete documents, access archive catalogs, etc.



The AT&T 3B2/300 computer with **HANDLE** on screen and integrated graphics printouts.



The first Office Automation System for UNIX.

HANDLE CORP., 140 MACKINAW ROAD, PO BOX 7018, TAHOE CITY, CA 95730 • 916-583-7283

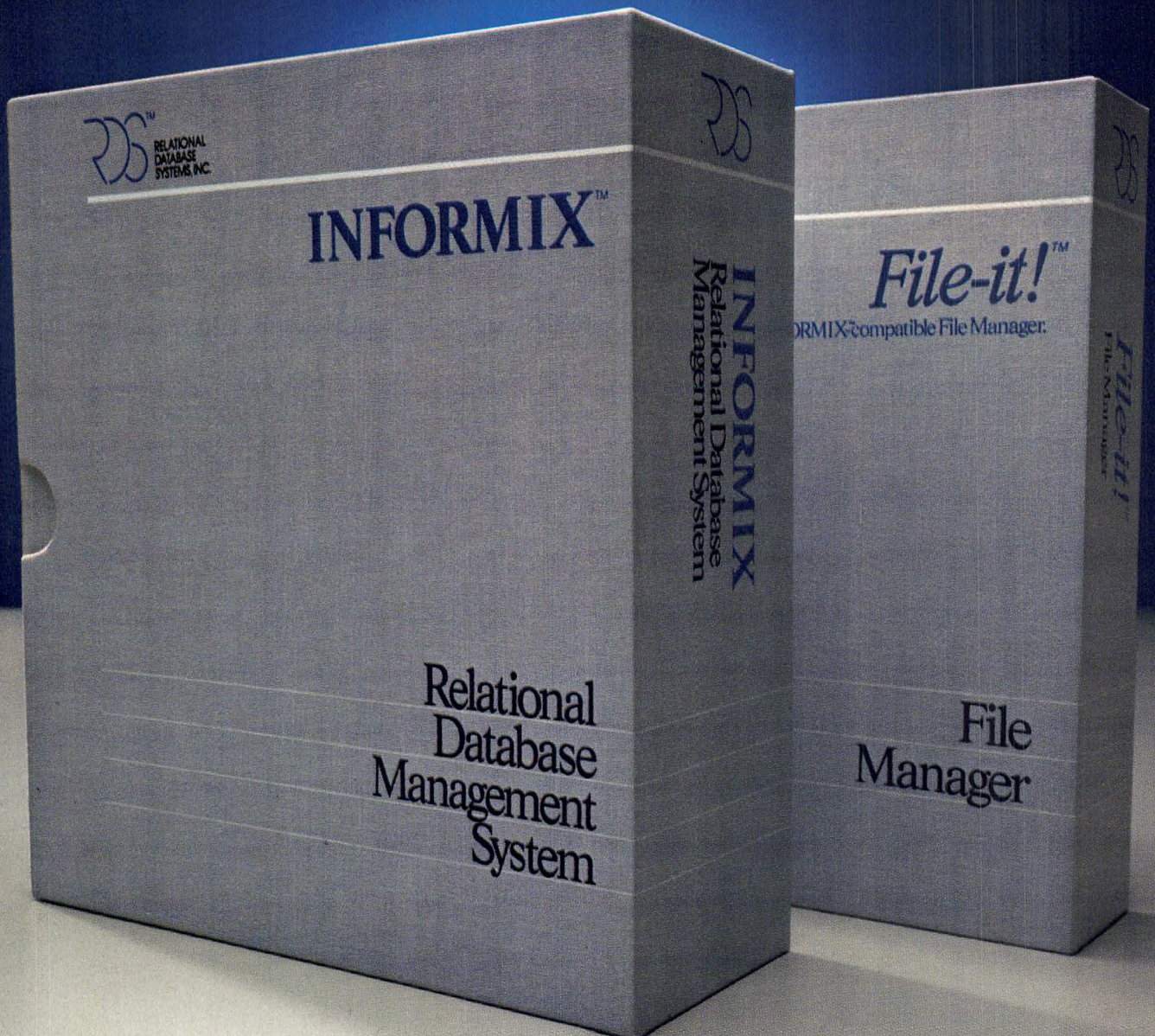
TM—HANDLE IS A TRADEMARK OF HANDLE CORP., 1983. UNIX IS A TRADEMARK OF BELL LABS. 3B2/300 AND 3B5 ARE TRADEMARKS OF AT&T.

**YOU CAN'T
PREDICT
THE FUTURE.**

**BUT
YOU CAN BE
PREPARED
FOR IT.**

WITH INFORMIX AND File-it!

**THE FIRST DATABASE SOFTWARE FAMILY
FOR UNIX AND MS-DOS.**



Now OEMs and systems integrators can sleep better at night. Because one company has taken the worry out of buying the right software.

RDS.

The company that produces a family of database software designed to take on the future.

Incompatibility is a thing of the past.

INFORMIX® and File-it!™ are compatible with UNIX™, MS™-DOS, PC-DOS™, and PC/IX systems (over 60 micros and minis* at last count).

INFORMIX is a true relational database system designed to take full advantage of the power of UNIX. It includes the most widely used report writer on the market.

Then there's File-it! The first easy-to-use UNIX file manager. Together, they have the flexibility to accommodate novices and experts alike.

INFORMIX and File-it! are fully integrated. Users can upgrade from File-it! to INFORMIX or access data from one program or the other without re-entering data, retraining employees or reprogramming.

Applications can also be moved from MS-DOS to UNIX and vice versa without having to rewrite the application.

Simplify program development.

RDS offers C-ISAM™, the de facto standard ISAM for UNIX. It's a library of C subroutines with a B⁺-Tree based access method that stores, retrieves and modifies data from indexed files. It's embedded in INFORMIX and File-it! Or is available as a standalone product.

Software good enough for AT&T.

AT&T, inventor of UNIX, has co-labeled INFORMIX, File-it! and C-ISAM to run on their full AT&T 3B Computer line (from micros to minis).

Hewlett-Packard, Altos, Zilog, Siemens, Cromemco, Perkin-Elmer, Sydis and General Automation have selected RDS as well.

In fact, INFORMIX has an installed base of over 6,000 copies. And RDS has sold over 35,000 licenses for all their products to date.

But before you make up your mind, check the facts one more time.

There's only one database software family that's UNIX-, PC-DOS-, MS-DOS- and PC/IX-based. It runs on more than 60 systems. And it's ideal for both novice and expert.

Now it doesn't matter where the future's headed. You're already there.

*RDS products are available for the following systems:

Altos 586, 986, 8600, 68000	Hewlett-Packard 9000
Apollo DN300	Series 200, 9000 Series 500
AT&T 3B2, 3B5, 3B20,	IBM PC, PC-XT
AT&T Personal Computer	Intel System 86/330
BBN C machine (all models)	Masscomp NC 500
Bunker Ramo Aladdin 20	Momentum Hawk 32
Charles River Data Systems	NCR Tower
Universe 68	Onyx C8002, C8002A
Convergent Technologies	Perkin-Elmer 3210, 3220,
Miniframe and Megaframe	3240, 3250
Corvus Systems Uniplex	Pixel 100/AP, 80 Supermicro
Cromemco System 1	Plexus P/25, P/35, P/40, P/60
DEC 11/23, 11/34, 11/44, 11/60,	Pyramid Technologies 90x
11/70, VAX 11/730, 11/750,	Radio Shack Model 16
11/780	SCI Systems IN/ix
Dual Systems System 83	Wicat System 150WS, 160,
ERG Mini System (all models)	200
Fortune 32:16	Zentec 2020
Forward Technology 320	Zilog System 8000
General Automation Zebra	(all models).
(all models)	

Demos of INFORMIX and File-it! are available. Demonstration software and complete manuals included.



**RELATIONAL
DATABASE
SYSTEMS, INC.**

**2471 East Bayshore Road, Suite 600, Palo Alto, California 94303
(415) 424-1300 TELEX 467687**

INFORMIX is a registered trademark of Relational Database Systems, Inc. RDS, File-it! and C-ISAM are trademarks of Relational Database Systems, Inc. UNIX is a trademark of AT&T Bell Laboratories. MS is a trademark of Microsoft and PC-DOS is a trademark of International Business Machines.



UNIXTM REVIEW

THE PUBLICATION FOR THE UNIX COMMUNITY

DEPARTMENTS:

- 6 Viewpoint**
- 8 Devil's Advocate**
by Stan Kelley-Bootle
- 14 The Human Factor**
by Richard Morin
- 66 Industry Insider**
by Mark G. Sobell
- 69 C Advisor**
by Bill Tuthill
- 82 /usr/lib**
by Jim Joyce
- 86 Rules of the Game**
by Glenn Groenewold
- 89 Off the Shelf**
by Doug Merritt
- 97 The Problem Solver**
by Bob Toxen
- 99 The UNIX Glossary**
by Steve Rosenthal
- 105 Calendar**
- 107 Advertiser's Index**

The cover illustration was created on a Digital Graphics Systems' workstation by artist Donna Cohen. The image was then transferred to film using the Rembrandt computer film recorder designed by Nise. The photo is courtesy of the Mark Twain Papers, Bancroft Library, University of California, Berkeley.

FEATURES:

20 TRICKS OF THE TRADE

by Mark G. Sobell

UNIX *can* make document production easier — honest!



20 A Rube Goldberg dream come true.

30 THE TROUBLE WITH TROFF

by Sandy Emerson

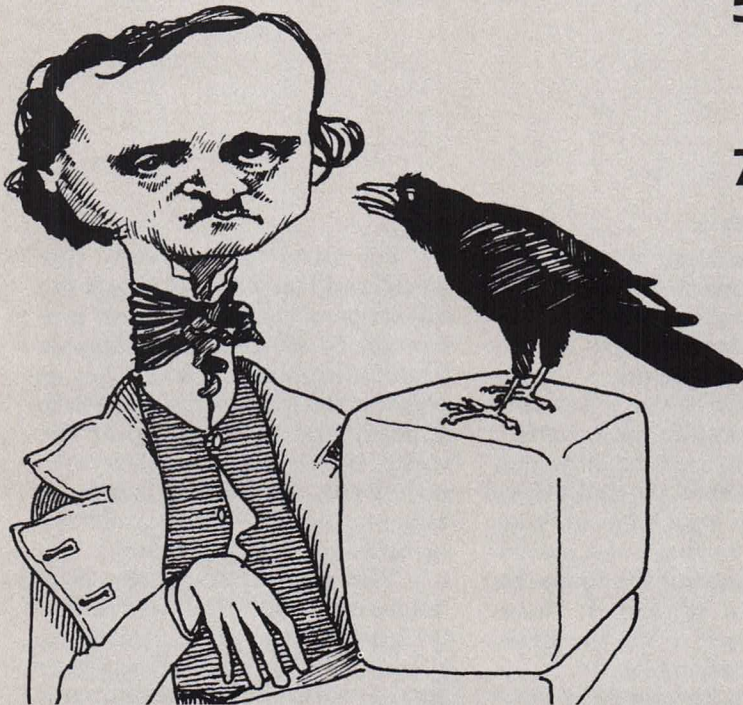
An admitted **troff** booster tells the awful truth.



58 The Joy of UNIX.

Volume 2
Number 5
August 1984

WRITING WITH UNIX



46 Cheer up! Writer's Workbench can help.

38 STRAIGHT TALK ABOUT MACROS

by Bill Tuthill

Comparisons between the four main macro packages are drawn by one who knows.

46 PROSE WITH STYLE

by Steve Rosenthal

Yes, Writer's Workbench offers you *style*. But will it suit your tastes?

56 PUTTING TEXT IN ITS PLACE

by Steve Rosenthal

An overview of the text formatting utilities offered in Documenter's Workbench.

58 INTERVIEW WITH BILL JOY

by Jim Joyce

The author of the C shell, *vi* and *termcap* talks about the text processing capabilities UNIX *should* offer.

74 VEXATIONS OF THE SEXIST FILTER

by Michael Donohue

The Writer's Workbench *sexist* program beats down even the most unrepentant.



74 The *sexist* program charges to the rescue.

77 THE 3B2 REPORT CARD

by Harry Avant and C. Andy Felong

AT&T supermicro is put through its paces.

UNIX REVIEW (ISSN 0742:3136) is published monthly by REVIEW Publications Co. It is a publication dedicated exclusively to the needs of the UNIX community. Subscriptions are \$28.00 per year (12 issues). Canada and Mexico add \$7/year. Overseas add \$20/year for surface mail. Address correspondence regarding editorial, press releases, product announcements to 520 Waller Street, San Francisco, CA 94117. Correspondence regarding subscriptions, change of address, USPS Form 2579, fulfillment and dealer sales should be addressed to 901 S. 3rd Street, Renton, WA 98055. Letters to UNIX REVIEW or its editors become the property of the magazine and are assumed intended for publication and may so be used. They should include the writer's full name, address, and home telephone. Letters may be edited for the purpose of clarity or space. Opinions expressed by the authors are not necessarily those of UNIX REVIEW. Entire contents copyright © 1984. All rights reserved and nothing may be reproduced in whole or in part without prior written permission from UNIX REVIEW. Editorial telephone: 415/621-6415. Subscription/Dealer Sales telephone: 206/271-9605. UNIX is the trademark of Bell Laboratories, Inc. UNIX REVIEW is not affiliated with Bell Laboratories. Application to mail at second class postage rate is pending at Renton, Washington 98055.

EDITOR:

Mark Compton

MANAGING EDITOR:

Pamela J. McKee

ASSOCIATE EDITORS:

Ken Roberts

Scott Robin

PRODUCTION DIRECTOR:

Nancy Jorgensen

PRODUCTION STAFF:

James Allen

Dan Arthur

Tom Burrill

Cynthia Grant

Corey Nelson

Florence O'Brien

Denise Wertzler

CIRCULATION

Tracey McKee

Barbara Perry

BUSINESS MANAGER:

Ron King

ADVERTISING**REPRESENTATIVES:**

Jules E. Thompson, Inc.

1290 Howard Avenue Suite 303

Burlingame, CA 94010

Lucille Dennis-415/348-8222

303/595-9299 - Colorado

312/726-6047 - Illinois

617/720-1888 - Massachusetts

713/731-2605 - Texas

Jules E. Thompson, Inc.

2560 Via Tejon

Palos Verdes Estates, CA 90274

Mary Winchell-213/378-8361

212/772-0933 - New York

TYPESETTING:

Data & Staff Service Company

PRINTING:

Thomas Ogle

EDITORIAL ADVISORS:

Dr. Stephen R. Bourne Software

Engineering Manager, Silicon

Graphics, Inc.

Jim Joyce, President, International

Technical Seminars, Inc.

REVIEW BOARD:

Dr. Greg Chesson, Technical Staff,

Silicon Graphics, Inc.

Ted Dolotta, Senior Vice President of

Technology, Interactive Systems

Corporation

Gene Dronek, Director of Software,

Aim Technologies

David Fiedler, President, InfoPro

Systems

George Goble, Systems Engineer,

Purdue University

Bill Joy, Vice President of Research

and Development, Sun

Microsystems

John Mashey, Software Engineering

Manager, Convergent

Technologies

Robert Mitze, Department Head,

UNIX Computing System

Development, AT&T Bell

Laboratories

Deborah Scherrer, Computer Scientist,

Mt. Xinu

Jeff Schriebman, President, UniSoft

Corporation

Otis Wilson, Manager, Software Sales

and Marketing, AT&T Technology

Systems

Walter Zintz, Executive Director,

Uni-Ops

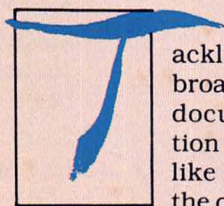
CONTRIBUTING EDITOR:

Ned Peirce, Systems Analyst, Bell

Laboratories

VIEWPOINT

The big plunge



ackling a theme as broad and deep as document production is very much like taking a dip in the ocean. It's simply not possible to swim the expanse.

As a result, we'll be staying a bit close to shore in this issue. Mind you, we won't be ducking any waves — we don't as a matter of policy. In setting the focus for this issue, we settled on those topics we found to be the most timely and challenging.

I hope you find many of your own questions answered. Why, for instance, has **troff** never really taken off? Can it truly be said that **vi** and **nroff** meet end user needs? What text processing alternatives are currently available to UNIX users? What sorts of future developments might we expect?

To answer these questions and others, we went straight to the experts and encouraged them to speak freely. They didn't need much prodding. Opinion, it seems, is in no short supply when it comes to UNIX document processing.

As you read through this issue, you'll find why Bill Joy, author of **vi**, has avoided the editor for years and why Sandy Emerson, an avowed **troff** "booster," now has a bone to pick. Other articles will tell you how to make the best of **vi** and **nroff**, where to find the strengths and weaknesses in four major macro packages and what to make of Writer's Workbench and Documenter's Workbench.

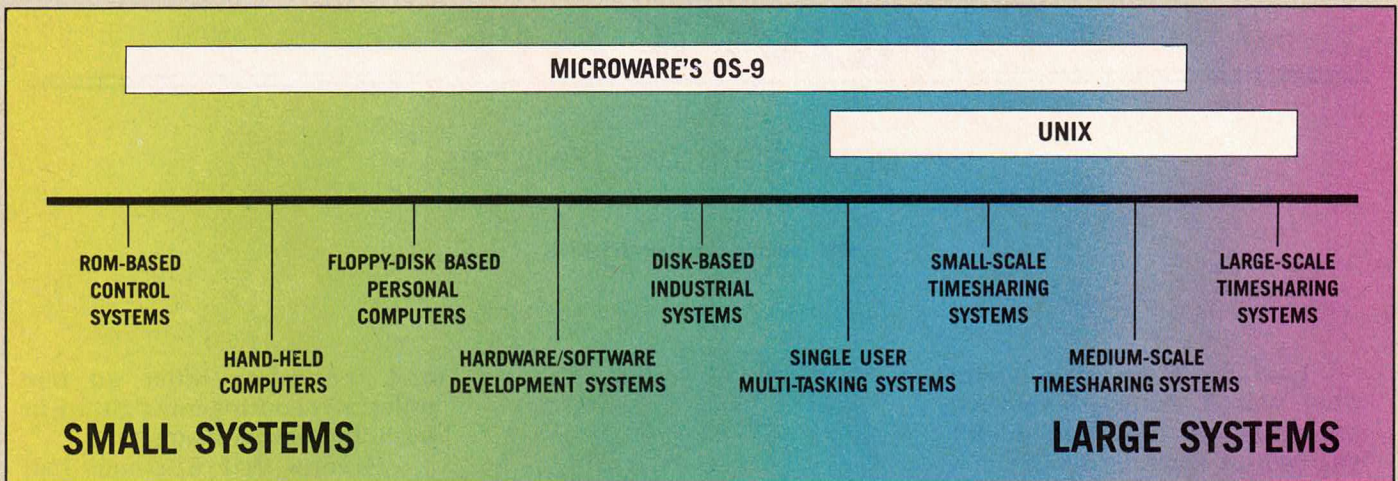
For all of that, though, this issue is still only a sampling of the full scope of UNIX document production. Space limitations keep us from addressing such utilities as **eqn** and **tbl** in anything more than a peripheral way. Many of the issues that go hand-in-hand with document production, such as office automation, also could not be satisfactorily addressed.

The topic, after all, is a substantial one. UNIX was initially funded as a word processing system for the Bell Laboratories legal department, and it retains to this day a word processing orientation in that all files are treated as text by the system. There's a lot of years, a lot of effort and, hence, a lot to say about UNIX document production.

In short, the topics we would have liked to explore and the authorities we would have liked to cite were many, but the pages were few. Thus, the last word on document production will not have been printed with the close of this issue. There will be future issues of *UNIX REVIEW* devoted to the subject, though they will not be released immediately.

There are, after all, a good many other topics that also demand our attention. It's a mighty sea we splash about in.

Only Microware's OS-9 Operating System Covers the Entire 68000 Spectrum



Is complicated software and expensive hardware keeping you back from Unix? Look into OS-9, the operating system from Microware that gives 68000 systems a Unix-style environment with much less overhead and complexity.

OS-9 is versatile, inexpensive, and delivers outstanding performance on any size system. The OS-9 executive is much smaller and far more efficient than Unix because it's written in fast, compact assembly language, making it ideal for critical real-time applications. OS-9 can run on a broad range of 8 to 32 bit systems based on the 68000 or 6809 family MPUs from ROM-based industrial controllers up to large multiuser systems.

OS-9'S OUTSTANDING C COMPILER IS YOUR BRIDGE TO UNIX

Microware's C compiler technology is another OS-9 advantage. The compiler produces extremely fast, compact, and ROMable code. You can easily develop and port system or application software back and forth to standard Unix systems. Cross-compiler versions for

VAX and PDP-11 make coordinated Unix/OS-9 software development a pleasure.

SUPPORT FOR MODULAR SOFTWARE — AN OS-9 EXCLUSIVE

Comprehensive support for modular software puts OS-9 a generation ahead of other operating systems. It multiplies programmer productivity and memory efficiency. Application software can be built from individually testable software modules including standard "library" modules. The modular structure lets you customize and reconfigure OS-9 for specific hardware easily and quickly.

A SYSTEM WITH A PROVEN TRACK RECORD

Once an underground classic, OS-9 is now a solid hit. Since 1980 OS-9 has been ported to over a hundred 6809 and 68000

systems under license to some of the biggest names in the business. OS-9 has been imbedded in numerous consumer, industrial, and OEM products, and is supported by many independent software suppliers.

Key OS-9 Features At A Glance

- Compact (16K) ROMable executive written in assembly language
- User "shell" and complete utility set written in C
- C-source code level compatibility with Unix
- Full Multitasking/multiuser capabilities
- Modular design - extremely easy to adapt, modify, or expand
- Unix-type tree structured file system
- Rugged "crash-proof" file structure with record locking
- Works well with floppy disk or ROM-based systems
- Uses hardware or software memory management
- High performance C, Pascal, Basic and Cobol compilers

microware[®]
OS-9[™]

MICROWARE SYSTEMS CORPORATION
1866 NW 114th Street
Des Moines, Iowa 50322
Phone 515-224-1929
Telex 910-520-2535

Microware Japan, Ltd
3-8-9 Baraki, Ichikawa City
Chiba 272-01, Japan
Phone 0473(28)4493
Telex 299-3122

OS-9 is a trademark of Microware and Motorola. Unix is a trademark of Bell Labs.

DEVIL'S ADVOCATE

Reports from the trenches

by Stan Kelly-Bootle

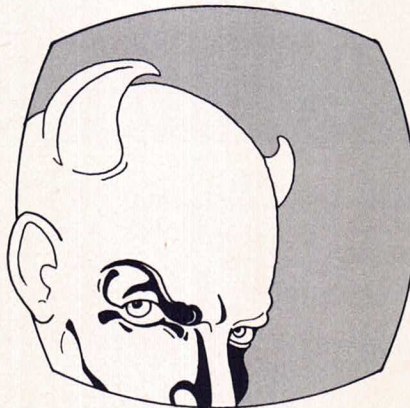
Last month we left you with Civil War II drifting somewhat aimlessly as IBM and AT&T engaged in sporadic sorties of a predominantly fiscal and propagandist persuasion.

Indeed, the conflict had become so bloodless and boring that ABC's "Monday Night Jihad" reverted to football, while the National Enquirer resumed its earlier lead story, "Siamese Twins' Sex-Switch Surgery."

AT&T had offered a 3B20D and two weeks in Maui with an inflight complimentary beverage, all for \$10 and proof of purchase labels from 15 family-size Del Monte fruit cans. IBM had retaliated by flooding the supermarkets with bogus labels (Spotter's Hint: the official labels have two c's and no accent acute in *macedoine*.) Worse still, IBM was dumping cut-price, rogue versions of UNIX with no vi editor. (Historians have noted a parallel in Thomas J. Watson's activities at NCR in 1903; see *IBM, Colossus in Transition*, Robert Sobel (Bantam Books, 1983) pp 38-39.)

A bemused public was then stormed by the AT&T Shell-of-the-Month Club, whose members suffered the regular, inertial threats typical of all such clubs:

"To enjoy the full benefits of System V, DO ABSOLUTELY NOTHING! However, if



you are dumb enough to wallow on in your present paludal environment, rush to the post office, get in line, buy a 20-cent stamp for our oversized card and mail it pronto. You have ten minutes starting NOW...

Students, having endured a decade without a decent protest-worthy cause, rallied instantly behind the Ma Belligerents with the ferocity of a bad case of acne. A successful "Bring Back Conscriptio" campaign at Kent State saw the first real casualties of the war, three draft-dodgers who were hung, tarred, drawn, feathered and quartered at a Joan Baez pro-war concert. Every campus had a "Back Our Persons at the Front" committee that knitted comforters and sharpened phone jacks for the Berkeley Irregulars, Vassar Volunteers, Stanford Light Foot

and the many other ad hoc university contingents fighting in far-flung foreign states.

Reports that Kissinger had been spotted commuting between Bell Labs and Armonk triggered fears that the contestants were about to settle out of camp, leading to renewed rallies and the symbolic burning of 80-column tab cards. At UC Davis, a 1401 was dragged into the square and dismembered by a jeering mob of exempted Divinity students, chanting "Butcher the Big Blue Bullies!"

The reactions of the business community, although less colorful and extroverted, nevertheless provided fascinating material for the scholars of transnational corporate politics. There were many old scores to be settled, many new profit windows to be opened, and much Machiavellian maneuvering to be done behind boardroom doors by companies wishing to be aligned with the eventual winner of the war. The emerging coalitions were far from obvious; some of IBM's erstwhile competitors boarded the Big Blue Bandwagon and switched their production lines with IBM's help from PC-compatibles to 3B look-alikes. AT&T, in turn, enjoyed succor from unexpected quarters as hundreds of disgruntled IBM VARs defiantly hoisted the AT&T Flag. IBM's own Tripoli Branch Office



With Alis™ everything works together. Text, spreadsheets, drawings, business graphics and database information work together in a single, always editable document.

Alis combines the advantages of integrated PC applications with the information-sharing benefits of communications-based OA systems. It offers the most advanced total office solution ever.

Alis makes it easy for people to work together. It provides integrated electronic mail, calendar and meeting scheduling, and revolutionary Automatic Office Assistants™ to aid management information monitoring and decision making.

Written in "C," Alis is initially available on UNIX* to large OEMs. It's destined to bring sanity to the mad tea party world of office automation.

*UNIX is a trademark of Bell Laboratories. Applix, Alis, and Automatic Office Assistants are trademarks of Applix, Inc.

Alis™

The next-generation office software system from **APPLIX**

Finally, some answers in Wonderland.

APPLIX, INC. 302 TURNPIKE ROAD, SOUTHBORO, MASSACHUSETTS 01772 (617) 481-4721

took advantage of the chaos and declared its independence, calling on all freedom-loving Product Centers to "Unite for an Everlasting Future and an Equitable Discount Policy!"

Wall Street was unsettled, to say the least, by a spate of unusual deals and mergers. Exxon, wagering that the two giants would be locked in a long war of mutual attrition, decided to strengthen its Office Products Division with a multi-billion dollar deal with Sperry, in which Exxon gave up its oil interests in exchange for the 1100 series. Sperry Petroleum quickly soared to record profits, and its slogan, "The Pump That Listens," was on everyone's lips.

The ailing Osborne Computer Corporation pulled off a remarkable greenmail coup by acquiring Texas Instruments. The new Osborne "Knapsack" model, with camouflaged screen and keyboard, offered Swiss-style neutrality by supporting neither UNIX nor PC-DOS and sold surprisingly well to both armies.

An offer of aid Hitachi made to IBM drew a rather guarded response. Eventually, a diplomatic compromise was reached under which Hitachi's Mishima Brigade, confined to strictly non-combat duties, toured the IBM camps with the Bob Hope Show, entertaining the troops with dashing displays of evisceration and decapitation. "There's Noh business like Show Business," as Hope never tired of quipping.

Burroughs promptly announced its solidarity with AT&T and sent Edsger Dijkstra to Bell Labs to support the war effort. In six days of divinely concentrated work, Dijkstra proved the incorrectness of all existing UNIX-based software, wrote a new 1K kernel in Algol84, produced a universal mandatory shell (immediately dubbed "The Royal Dutch") and sketched the designs

for a 3B7700 series. On the seventh day he rested on a flight back to Amsterdam.

Kernighan and Richie, far from happy at finding the C language so brusquely superceded, manfully accepted the new order in a Dunkirk display of unity. After quietly adding structure variable pointers to Algol84,

**Students,
having endured
a decade without a
decent protest-worthy
cause, rallied
instantly.**

they joined Thompson, Bourne, Feldman and the rest of the team in a feverish effort to rewrite all the outmoded UNIX utilities. All agreed that NUNIX, as the new system came to be known, was undoubtedly *correct* from a theoretical standpoint; yet the complete loss of *portability* was a tough pill to swallow. NUNIX was running, or rather limping, but only on the hastily built 3B7700 prototype. IBM seemed to have cornered the IC market, and even Fairchild reject chips were hard to find. The 3B7700 was kept alive with bits of cannibalized ILLIACs and daring guerrilla raids on the Boston Computer Museum. The maintenance team, reinforced by Los Bochos, a group of auto mechanics from Tijuana, performed nightly miracles of improvisatory rectification. These were, indeed, times to try the souls of men, but as Rob Pike remarked, "You can't make

omelettes without breaking a few shells."

An early victim of the war, of course, was the "open architecture" philosophy, as both sides took rigorous steps to safeguard vital trade secrets. Spies disguised as kibbitzers lurked at every CRT. The Ma Atari Mole College enjoyed all-time record enrollments. The peacetime price of a superuser password doubled overnight. "Careless Coding Costs Lives!" warned posters at every installation. Caveat preambles to important source listings were redrafted by Glenn Groenewold for added protection. Gone was the informal warning:

"I say, old chap, I'd really much rather you didn't copy this program without checking with me first, you know, fair's fair and all that..."

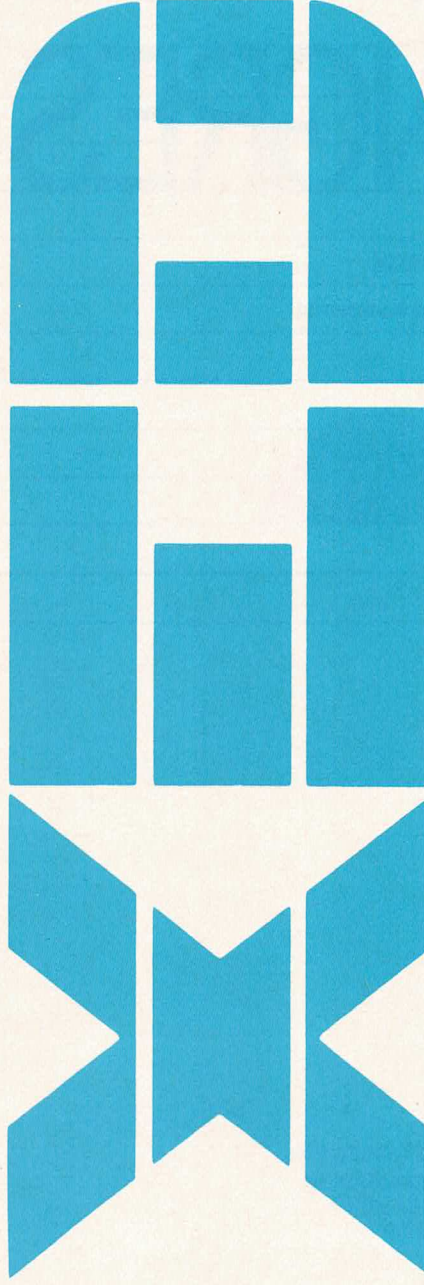
and in its stead was placed the blood-curdling "Macbeth-Flynt Clause," which, alas, cannot be fully revealed in a family magazine such as this. My Liverpudlian sensitivity forces me to * the unpleasant words in this brief extract:

"Be hereby warned that * *
***** smegma
***** mother * * *
garrote * * toad * gibbet."

Connector and chip pins were regularly and arbitrarily renumbered, low/high byte conventions randomly reversed and previously unemployable technical writers hired to produce confusing documentation. (Readers who have dared to doubt my reportage are invited to study the evidence.) The resulting nightmare of obfuscation affected friend and foe alike, but as Richard Gauthier put it, "Better that 60 allies be confused than that one enemy receive enlightenment."

The technical spinoffs of previous wars have been widely

CMC COMPUTER METHODS LIMITED



DOCUMENT PROCESSING SYSTEMS

- Interactive Spelling
- Balance Sheet Mathematics
- Auto Footnoting
- Auto Outlining
- Phrase Glossary
- Template Assembly
- Photo Typesetter Interface
- Widow-Orphan Elimination
- Pagination
- Form Generation
- Electronic Mail
- Data Entry
- Financial Analysis

Box 709 Chatsworth, CA 91311 (818) 884-2000 TWX910-494-1716 Int'l Tlx 292-662 XED UR

documented (see, "Pro Bello," Pope Boniface III; and "In Praise of Carnage," Beatrix Potter (Faun Books)); the Phangisars did much to improve the tensile strength of silk; the Wars of the Roses brought us the first pesticides further refined in Vietnam); Austerlitz introduced the Steak Tartar, while corned beef and the safety match serve to remind us

that Verdun was not in vain. Civil War II will undoubtedly be remembered for its contributions in the field of computer security.

IBM set the pace by fitting all 3270s with PC Jr. keyboards, thus, restricting access and solving an acute inventory problem in one master stroke.

AT&T countered with the "Simple Simon" protocol that

allowed all commands to be blocked unless preceded by **simonsays**, or its mnemonic **ny**.

Jim Joyce's three-volume *Simon By Example* explains the origins (he got the idea while listening to kids playing in a New York park) and lists the following paradigms:

```
$ cat topsecret
cat: cannot open topsecret! missing simonsays
```

```
$ ny cat topsecret or $ simonsays cat topsecret
We plan to attack at dawn. Check your mail
for details.
```

```
$ ny cat topsecret > myfile
cannot redirect without simonsays!
```

```
$ ny cat topsecret ny > myfile
[myfile now holds topsecret]
```

Use of yn to switch off protection:

```
$ yn cat topsecret
cat: cannot open topsecret! missing simonsays
```

```
$ ny yn cat topsecret
We plan to attack at dawn.
```

Volume II of "Simon By Example" covers the Berkeley variants, the chief of which is the cynical **dreyfussays** or **dl** option, controlled thus:

```
$ ny whosays
simon
```

```
$ ny whosays dreyfus
```

```
$ ny whosays
cannot say! missing dreyfussays
```

```
$ dl whosays
dreyfus
```

The "Simple Simon" play was remarkably successful: dazed IBM spies deserted in droves and confusion.

Next month: No surrender!

Stan Kelly-Bootle is a grizzled mainframer who worked on the pioneer EDSAC I at Cambridge University in the early 1950s. As founder/President of the LISA Moaners' Club, he urges more machismo and less user-mollycoddling in software. In spite of some reservations, he feels that UNIX is a bandwagon heading in the right direction. His exposé of computer scientific epistemology in the lexicographic environment, "The Devil's DP Dictionary" (McGraw-Hill, 1981), is to appear soon in a Japanese language edition (Shizen Sha, Tokyo). ■

INGRES



**The Relational
Database Management
System**

- Powerful/Easy to Use
- Visual Programming Capability with:
Application Generator
Integrated Report Writer
Integrated Business Graphics
- Integrated Data Dictionary
- Non-Procedural Query Language
- Exclusive Networking Capability
- Supported Around the Clock

**AT&T INGRES is now available on
the 3B5 and AT&T INGRES/CS on
the 3B2!**

VAX and VMS are trademarks of Digital Equipment Corp.
MC68000 is a trademark of Motorola Corp.
UNIX is a trademark of Bell Laboratories.

Relational Technology Inc.
2855 Telegraph Avenue
Berkeley, CA 94705
415-845-1700

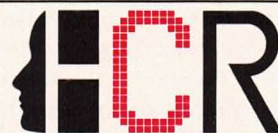
Circle No. 271 on Inquiry Card

UNIX Operating Systems are becoming the versatile, powerful backbones of more and more computer systems. Helping people build the many advantages of the tried-and-true UNIX[™] Operating System into their systems is our business. Our UNITY[™] adaptation of Bell Laboratories UNIX Operating System is available for the DEC PDP-11[™], and for the DEC VAX[™] both under VMS[™] and stand-alone. We also have UNITY for systems based on the National Semiconductor 16032 and the Motorola 68000 available for OEMs.

AND our growing array of software based on the UNIX Operating System includes all the major languages; for example C, FORTRAN, PASCAL and BASIC. We also have packages for word processing, typesetting, and relational database management; an electronic worksheet; and a user friendly menu shell.

YOU can build these benefits into your system with full confidence, because we also back up all our software products with full customer training and support. Hundreds of customers around the world are now using HCR software products. To find out how we can put you and the UNIX Operating System together, call or write:

Human
Computing
Resources
Corporation



10 St. Mary Street, Toronto, Ontario, Canada M4Y 1P9 (416) 922-1937

Circle No. 270 on Inquiry Card

THE HUMAN FACTOR

A smorgasbord of UNIX text processing utilities

by Richard Morin

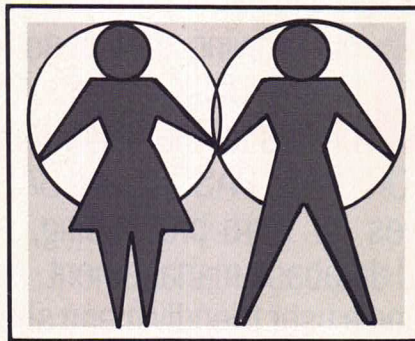
The UNIX approach to document production is entirely consistent with the operating system's overall design philosophy. Experts are allowed, even encouraged, to kludge solutions to text processing problems while others are left to cast about more or less aimlessly among hundreds of possible alternatives.

Nowhere in UNIX does one find a specialized program posing as a complete "document production system." Instead, the system supplies a number of helpful commands that can be used either alone or in combination to perform desired document production tasks. These can be used in either an ad hoc or controlled manner.

Many commands have something to do with text manipulation and/or document production, since these have long been UNIX design goals. In fact, UNIX was originally funded as a word processing facility for the Bell Laboratories legal department.

UNIX regards almost every file as a textfile so some of the most useful commands have very wide ranges of utility. These ranges are not always adequately covered by the growing number of tutorials devoted to UNIX document production utilities.

In an attempt to give a brief overview and handy reference to the UNIX document production



capabilities, this column will categorize the most relevant commands. Some commands that I mention may not exist in a given UNIX implementation, and some useful commands that do exist may not be listed here. The source for the following list is the 4.2 BSD implementation produced by Sun Microsystems.

The main UNIX document production utilities are text editors, formatters and preprocessors. They are as powerful and mysterious as any devout UNIX hacker might wish. The "vi Quick Reference" card lists well over 100 control sequences. The formatters and preprocessors also use numerous arcane codes and mnemonics.

It is quite possible to use these programs knowing only a small subset of the command repertoires. Unfortunately, UNIX documentation does not encourage this. Instead, it tends to

assume that every user will wish to know about every possible feature. Let me, therefore, encourage you to look to tutorials for help in finding a good starting subset.

The line-oriented editor **ed** is found on all UNIX systems, and is thus good to know at least marginally. The batch version of **ed** known as **sed** can be very useful in performing large and/or repetitive editing tasks. The Berkeley **vi** editor is screen oriented and is becoming available on an increasing number of systems. It is also much easier to use than **ed**.

A "safe" version of **vi** named **view** is available for viewing and/or modifying files without risking damage to original files. In addition, a line-oriented version of **vi** named **ex** is available. Proprietary editors are now also becoming available under UNIX. Perhaps the most interesting of these is **emacs**, which supports multiple editing windows and a subset of **lisp** (!) as part of its repertoire.

Despite the fact that UNIX is generally designed for interactive use, its document production facilities are quite batch-oriented. UNIX tools assume that the user will produce a piece of text – adding appropriate control codes along the way – run it through formatting utilities and then

UNIX Operating Systems provide the standard environment for applications so they can be ported across advanced multi-tasking and multi-user systems. Word processing, spreadsheets, and database management are some of the applications already available under UNIX. Until now, no business and financial software has been available. Until now, the only solution has been to retrofit existing financial packages — packages that were written for systems with limited storage resources and limited interactive capability. Until HCR's Advanced Business Applications Software for UNIX.

BUSINESS software designed especially for the UNIX environment is now available from Human Computing Resources. Incorporating an ever-growing series of integrated modules with a powerful relational DBMS, the package presently includes Accounts Payable, Accounts Receivable, and General Ledger in a form that users can query and interact with directly. HCR's software takes advantage of System V, making it the first truly multi-user system.

SOFTWARE like this could only come from HCR. This new applications family is designed so OEMs, distributors and dealers can easily support, extend and adapt it. Demonstration versions are now being evaluated by OEMs. To find out how we can put our business software into your UNIX system, call or write:

Human
Computing
Resources
Corporation



10 St. Mary Street, Toronto, Ontario, Canada M4Y 1P9 (416) 922-1937

examine the results. The batch approach works well when the experienced user can predict the effects of a given decision. It is less appropriate in situations where either the user is a novice or the desired effects have not been established.

Contrast this with the highly interactive nature of the Macintosh or Interleaf systems, or even screen-oriented text editors available for other operating systems. These interactive systems provide the user with immediate feedback, allowing a more experimental approach to

UNIX regards almost every file as a text file so some of the most useful commands have very wide ranges of utility.

the job at hand. Fortunately, interactive text formatting programs are becoming available to UNIX users. Interleaf, in fact, is currently available for use on Sun workstations, although its price (approximately \$8000) may discourage casual purchase.

The **nroff** and **troff** text formatters are quite useful in preparing documents which must conform to a standardized style. Since both are too difficult for direct use by most people, they are generally used in conjunction with the **me**, **mm** or **ms** libraries of formatting macros. This column is being written in **ms** format, and will be sent to the editor in that

form. Upon receiving it, he will be able to edit the text as needed, using **nroff** to produce inspection copies.

The preprocessors **eqn**, **neqn**, **refer**, **tbl** and **vgrind** produce formatting commands from descriptions of equations, bibliographic reference databases, descriptions of tables and program textfiles respectively. Finally, the **checkeq**, **checknr**, **col**, **colcrt**, **deroff**, **pti**, **soelim**, **ul** and **vfont** utilities are an assortment of (sometimes) useful tools for use with **nroff** and **troff**.

The utilities **addbib**, **indxbib**, **refer**, **roffbib** and **sortbib** are particularly useful to authors of scholarly papers. These programs maintain and access bibliographic databases, assisting with footnote preparation, citation searching, and other measures common to academic writing. The user may draw upon either a privately maintained database of references or a supplied database containing over 4000 items.

There are some tasks for which powerful formatting utilities are simply too fancy. The **colrm**, **fmt**, **fold** and **pr** utilities can help in such situations. They provide simpler and/or more specialized capabilities than those found in the **nroff** and **troff** family.

While the Source Code Control System is not usually considered to be a document production tool, it can be extremely valuable in tracking changes made to a set of documents. The **SCCS** family of routines includes **admin**, **cdc**, **comb**, **delta**, **help**, **get**, **prs**, **rm del**, **sact**, **scs**, **scsdiff**, **scsfile**, **unget**, **val** and **what**. Similarly, the **make** program can be very useful in coordinating the use of other utilities, insuring that all necessary steps are taken following a change to a document.

Any serious effort at document production will produce, if nothing else, a copious amount of stored text. Since much of this text is relatively inactive at any given instant, it may be profitably compressed. For files containing large numbers of blanks, the **unexpand** command can obtain a considerable savings, converting strings of blanks into appropriate numbers of tabs. The more generally applicable **ccat**, **compact** and **uncompact** utilities use an adaptive Huffman algorithm, achieving a typical compression of 38 percent from the original text.

The UNIX protection scheme, when used properly, is generally sufficient to insure the privacy of sensitive material. Occasionally, however, text must sometimes be sent through unprotected channels. The **crypt** utility is a relatively secure way of protecting such text. It should be noted that the US government prohibits the shipment of **crypt** outside the US.

It is often useful to know whether two textfiles differ, and if so, where and how. The **cmp**, **comm**, **diff** and **diff3** commands provide various answers to this need. UNIX makes it very easy to ask the computer to do comparisons and calculations which would be time-consuming and tedious to perform manually. These simple commands can save a great deal of hassle if used properly.

Sometimes a simple look at a file can be very informative. Aside from **view**, which was mentioned earlier, UNIX provides such commands as **cat**, **head**, **more** and **tail**. Another program, the infamous **grep**, is useful for viewing a specified subset of a file's contents. Based on the **ed** "global (search for) regular expression & print" command (**g/.../p**), **grep** will cheerfully scan any number of files, looking for specified text

WHY DEC AND INTEL CHOSE THE MARK WILLIAMS C-COMPILER.

DEC and INTEL wanted the best C technology available, with excellent code density, supporting the full C language and their specific operating environments—all at a competitive price.

They found it all at Mark Williams.

WHY YOU SHOULD CHOOSE THE MARK WILLIAMS C-COMPILER.

Our C-compiler supports the dominant 16-bit microcomputers—68000, PDP-11, Z8000, 8086—with a proven reliable, high-technology product. We are shipping versions of C for a large number of environments including CP/M and PC DOS. Both cross and native compilers are available.

Call us for the distributor nearest you. OEM's should contact us directly about their specific requirements.

Mark Williams Company,
1430 West Wrightwood, Chicago, Illinois 60614,
312/472-6659.



strings. Each matching line will then be sent to standard output.

Character and text string substitution capability can also be useful. The **expand** and **unexpand** commands translate between blanks and tabs, and the **tr** utility does much more general character substitution. The stream editor, **sed**, which was described above, can do even more powerful substitutions.

```
<<List the file.>>
% cat text
A man, a plan, a canal, panama.
Able was I ere I saw Elba.

<<Convert it to lower case.>>
% tr A-Z a-z < text
a man, a plan, a canal, panama.
able was i ere i saw elba.

<<Make a word list, 1 word per line.>>
<<Non-alpha characters are changed>>
<<into single newlines.>>
% !! | tr a-z -cs ' 12'
a
man
a
plan ^C

<<Bring identical words together.>>
% !! | sort
a
a
a
able ^C

<<Count repeated words.>>
% !! | uniq -c
  3 a
  1 able
  1 canal ^C

<<Sort to decreasing order of use.>>
% !! | sort -rn
  3 a
  2 i
  1 was
  1 saw ^C

<<Save the last few commands typed.>>
% history > wf
```

Figure 1

Finally, the **m4** macro processor, while not for the faint of heart, can perform a truly remarkable range of operations.

The **awk** pattern scanning and processing language is modeled after C, but is much easier to use, being interpretive rather than compiled and less restrictive in its variable usage. It can be used in a manner similar to the translation commands mentioned above, or in other ways altogether. Ignored by many UNIX users, it is in fact one of the most convenient and powerful text manipulation tools available.

The appearance and readability of a document depends greatly on its spelling, style and diction. The **spell** and **look** commands aid in detecting and correcting spelling errors. Meanwhile, the **diction**, **explain** and **style** utilities, part of the Writer's Workbench package, perform surface analyses of English usage.

When producing a document that must be a certain length, the **wc** command for counting characters, words and lines can be very useful. Note that **nroff** formatting commands will also be counted in the total unless they are removed first by piping them through **deroff**, as in the following:

```
%deroff file | wc
```

The ability to string together sequences of commands as filters is one of the most powerful features of UNIX. Combined with the C shell's capabilities for incremental development and testing of commands, this feature allows users to build powerful commands with a trivial amount of effort.

Let's assume, for instance, that there is a need to perform a simple word frequency analysis on textfiles to help detect the overuse of words. The program

would need to display, a screen at a time, a list of the words used in a document, along with the number of times it appears, sorting in decreasing order of use.

Such a routine can be built up and tested in an incremental manner. Note that in the **cs** example in Figure 1, **!!** re-executes the previous command, **^C** is being used to kill a running command and **history** is used to capture the text of the final commands.

After a bit of editing, the new **wf** command can be saved in the following way:

```
tr A-Z a-z < $* | (
tr -cs a-z ' 12' | (
sort | (
uniq -c | (
sort -rn | (
more
```

An attempt to write this sort of program in FORTRAN or COBOL (!) would consume a substantial length of time. But the script above can be written on a single line, and it took only a minimal amount of time to compose. This kind of flexibility and convenience has been noted as the hallmark of UNIX programming, but it obviously also extends to UNIX document production.

The commands described above, along with any new ones your own undoubtedly fertile imagination can conjure, should provide a good set of document production tools. Why not put them to good use?

Richard Morin is an independent computer consultant specializing in the design, development and documentation of software for engineering, scientific and operating systems applications. He currently operates the Canta Forda Computer Lab in Ft. Washington, Maryland.

Gould...Innovation and Quality in UNIX-based systems

Our Firebreathers are scorching old performance standards.

Gould's PowerNode™ 9000 blasts through UNIX* benchmarks at 4.5 times the speed of the VAX™ 11/780. Sound impossible? Give us your real production code or benchmarks and let us prove it.

Firebreathing Performance.

Now you can run software development and production at the same time, with highly responsive performance. Tightly coupled dual processors nearly double throughput and virtual memory accommodates large programs. Hardware fixed point and floating point accelerators retain high performance in heavy number-crunching situations. The PN9000 handles mainframe jobs in a multi-user UNIX system or serves as a backend processor in a widely distributed network.

Unique UNIX Software.

Gould's own high performance UNIX-based operating system (UTX/32™)—a unique combination of Berkeley 4.2 with special Bell System V features—makes it easy for you to use your VAX-based UNIX software. This allows easy conversion from your system to the increased power of a Firebreather.

Compatible Family.

Gould's Compatibility Suite is a collection of application software packages that are compatible across the entire PowerSeries™ product line. Use C, Cobol, BASIC, or Pascal languages intermixed. This close-knit processor family offers all the advantages of

a dedicated system plus the lower-cost-per-user option of sharing resources with Gould's standard networking capabilities including Ethernet™. The Firebreathers are the high end of the widest range of UNIX-based systems in the industry.

Gould's Firebreathers are scorching the UNIX market.

**Gould Inc.,
Computer Systems Division**
Distributed Systems Operation
6901 West Sunrise Boulevard
Ft. Lauderdale, Florida 33313
(305) 797-5459

*UNIX is a trademark of AT&T Bell Labs
™PowerNode, PowerSeries and UTX/32
are trademarks of Gould Inc.

™Ethernet is a trademark of Xerox Corporation

™VAX is a trademark of Digital Equipment Corporation



GOULD
Electronics

TRICKS OF THE TRADE

Making document production easier
and more effective

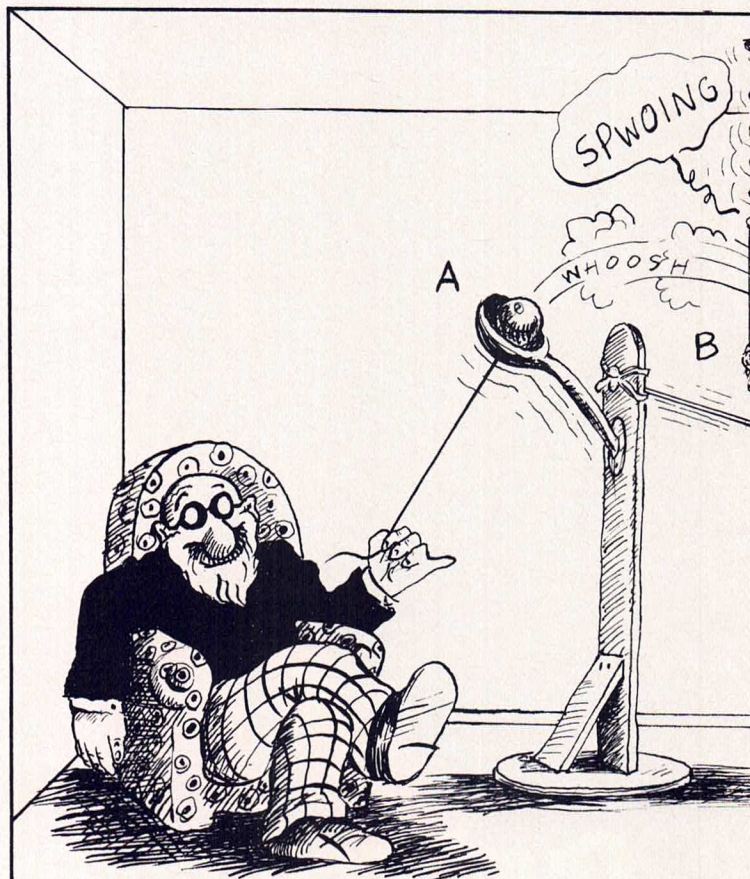
by Mark G. Sobell

Document production starts with an idea you want to communicate in writing. It continues as you record your thoughts on magnetic media or paper, and concludes when you have a document that serves the needs of your audience. The document you produce can be anything from a handwritten note to a piece of electronic mail to a typeset book.

This article discusses ways in which you can use standard UNIX utilities (**vi** and **nroff**, mostly) to reduce the work you put into a document while maintaining or improving its effectiveness.

MAKING THE COMPUTER DO THE WORK

You can take advantage of your computer's power while you are writing. A word processing system should do more for you than register keystrokes – a typewriter can do that. A text editor allows you to try out different ideas with a minimum

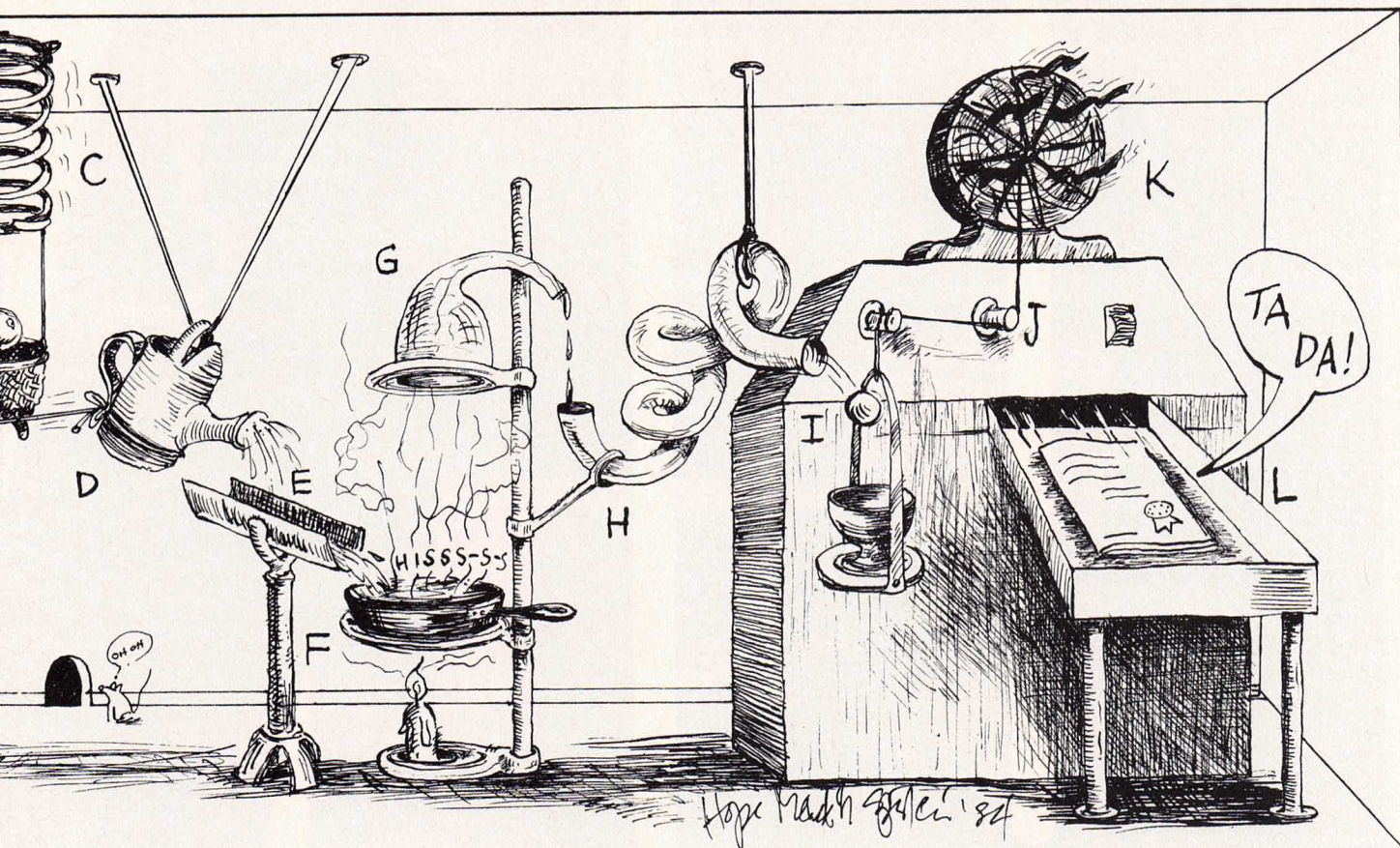


USER INPUT CAUSES DATA TO BE
COMPILED (F), FED BY FRESH
INTO FILE (I), WHERE SHELL S

of effort. You can move a phrase, sentence or paragraph and see how it looks and feels. If you don't like the results, you can move it back.

You can develop your own style for working with a machine. As an example, I compose my thoughts at my terminal, allowing myself to put down ideas, even if they are not in their final polished form. As I think, I write, allowing my whimsy to dictate to my fingers. When I see a way to improve a sentence or phrase, I make a copy of the sentence just after the original and then alter the copy. Sometimes the changes don't help and I delete the changed sentence. In the event I like the changes, I delete the original. If I can't make up my mind, I keep them both and delete one or the other later.

Comments. If you are using **nroff** to format your text, you can embed comments in the text to remind yourself of points you want to make, areas that need cleaning up or places in the text that need more work



PROJECTED TO FILE (B) TRIGGERING DATA TRICKLE (E).
DATA, CAUSES PARSER (G) TO SEND INPUT THROUGH FILTER (H)
SCRIPT (K) PRODUCES FINISHED DOCUMENT (L).

Illustration by Hope Epstein

for any reason. The comments will not appear when **nroff** processes the document, but they are there for your use while you are editing.

A comment begins with a `\"` if it occupies a line by itself and `\"` if it follows text on a line. The end of a line terminates all comments. The following samples show both kinds of comments:

```
\" This comment occupies an entire line.  
This is text that \" Here's a comment that  
nroff processes. \" follows text on a line.
```

Easy Editing. Using an editor such as **vi**, you can store your text in such a way as to make later manipulations easier. If you begin each sentence on a new line, it will be easier to move sentences around in subsequent drafts. Also, you frequently edit the beginning or end of a sentence. If you break your sentences into component phrases, you can edit your text more efficiently. Although not a hard and fast

rule, it can be convenient to break lines after commas or other punctuation, especially if they occur near the end of a line. You can see at a glance what is in a list if you put each item in the list on a separate line.

Put your text into the computer in a format that is convenient for what you are doing: writing and editing. Let the computer and **nroff** do the work of formatting.

Spaces After Periods. Good style requires that two spaces follow a period that ends a sentence. There are two ways to cause **nroff** to put two spaces after a period. You can put the period at the end of a line and let **nroff** take care of inserting two spaces, or you can manually put the two spaces after the period if it appears in the middle of a line. Following the theory of having the computer do the work and setting up your file to make editing easier, putting the period at the end of the line is the preferable method.



UNIX[™] operating systems. An ideal has been

If you've been waiting for an ideal operating system, your wait is over. Now there's HP-UX. It is Hewlett-Packard's enhanced version of the industry-standard UNIX operating systems. And it's available right now on a wide range of HP computer systems.

Yes. It's running on our MC68000-based machines and our powerful 32-bit systems, so

you can pick the right computer for the job.

There are extra features such as graphics and networking. Plus there's a growing array of applications software available for you to take advantage of.

And the HP-UX operating system is backed by our full service organization. As with each of our high-powered systems, we're ready



realized.

to answer questions. Working with both end-users and OEMs, we'll find the best solution for any particular application.

Sound interesting? Call your local HP sales office right now about the HP-UX operating system. Or write to Hewlett-Packard, Attn. Pat Welch, Dept. 100194, 19447 Pruneridge Ave., Cupertino, CA 95014. In Europe, con-

tact Henk van Lammeren, Hewlett-Packard, Nederland B.V., Dept. 100194, P.O. Box 529, 1180 AM Amstelveen, The Netherlands.

Productivity. Not promises.



**HEWLETT
PACKARD**

BDO2421
UNIX is a trademark of AT&T Bell Laboratories.

Circle No. 39 on Inquiry Card

If **nroff** isn't leaving two spaces when you have a period at the end of a line, check that there is not a space after the period. A space following the period will cause **nroff** to leave only one space.

To see if there is a space at the end of the line, move the cursor to the line in question and, with **vi** in command mode, give the command **:l** (for list). (This and all other colon commands move the cursor to the status (bottom) line of the screen. You must terminate all colon commands by pressing RETURN.) When you press RETURN, **vi** displays the current line with a dollar sign marking the end of the line. A space between the period and the dollar sign indicates that there is a space at the end of the line. Remove it and **nroff** will leave two spaces when it processes the text.

Use the **:set list** command if you want **vi** to display a dollar sign at the end of each line of text. These dollar signs are not part of the text and are visible only while you are using **vi**. (Use **:set nolist** to remove the dollar signs — see "Setting Up Your Terminal" later in this article for more information.) The **:l** and **:set list** commands are also useful for tracking down elusive tabs. These commands display tabs as **^I**.

Removing Trailing Spaces. You can get rid of all trailing spaces with the following command. (You may want to write out the file with **:w** before you try it.):

```
:%s/ *$//
```

This substitute command replaces trailing spaces on all lines with nothing. The **%** represents all lines in the file. If **%** doesn't work on your system, use **1,\$** in its place. The **s** (substitute) command replaces the characters between the first and second slashes with the characters between the second and third slashes.

The key to this command is the regular expression **"*\$"**. Within a regular expression, an asterisk represents a group of zero or more characters and a dollar sign represents the end of a line. The string **"*\$"** therefore matches zero or more spaces at the end of a line. The substitute command replaces each occurrence of zero or more spaces at the end of a line with nothing (**//**).

SPEEDING UP EDITING

The Period Command. The period command repeats the last command you gave **vi** that altered the text. If you just inserted a line of text, it inserts the same line again; if you just deleted a word, it deletes another. This command is useful when you want to change some, but not all, occurrences of a word in a document.

Suppose you want to change most occurrences

of "chapter" to "section." Search for the next appearance of "chapter" in the file with a **/chapter** command. (Like colon commands, the search commands **/** and **?** move the cursor to the status line and must be followed by a RETURN.) After **vi** finds the first occurrence of "chapter," change it by giving the command **cw** (for change word), enter "section" and press ESC. Press **n** (for next) to find the next occurrence of the same word. You can now press **.** (period) to repeat the last command you gave that changed text. The editor changes "chapter" to "section" without requiring you to enter either word. Press **n** and **.** again when you want to repeat the substitution. When you finally get to a "chapter" that you don't want to change, don't give the period command. Just press **n** to find the next occurrence of "chapter."

Using this technique, you can breeze through a document with one finger on the **n** key and one on the period key, changing only some of the words **vi** finds.

Backing Up the Cursor. Another handy technique for speeding up writing and editing is the use of CTRL-W to back up over words while you are using **vi** to enter text (in insert mode). If you notice a

**A word processing system should
do more for you than register
keystrokes - a typewriter can do
that.**

mistake a couple of words back as you are entering text, press CTRL-W repeatedly to move the cursor back a word at a time. Then enter the correct word and continue. CTRL-W is similar to CTRL-H (the default backspace) except that it backs up by words instead of by characters. It will not back up past the beginning of the line or past the text you have entered most recently. As with CTRL-H, CTRL-W does not always erase words from the screen as it backs up over them, even though it removes the words from the text you are editing.

Along the same lines is the backup-to-the-beginning-of-the-line key. This is the same key the shell recognizes as your order to backup to the beginning of a command line. The standard key used for this purpose is **@**, but it varies from system to

st portable Manx Software Systems
with over 6,000 licensed
C users is one of the leading suppliers of
"C" compilers and cross compilers for
the world's most popular micro computers. The
MANX AZTEC C compiler is available as a cross
native compiler for the following environments:

PC DOS
MS DOS
CP/M-86

CP/M-80
TRSDOS

APPLE DOS
COMMODORE 64 (12/83)
ProDOS (2/84)

CROSS DEVELOPMENT SYSTEMS

HOSTS

PDP-11	68000	8086/8088	8080/8085/Z80
UNIX	UNIX ports	UNIX ports PC DOS MS DOS CP/M-86	CP/M-80

TARGETS

8086/8088	8080/8085/Z80	6502/65xx
PC DOS	CP/M-80	APPLE DOS
MS DOS	TRSDOS (12/83)	APPLE ProDOS (2/84)
CP/M-86	LDOS (12/83)	COMMODORE 64
	DOSPLUS (12/83)	

NATIVE COMPILERS

For prices and information call:
800-221-0440 (outside NJ)
201-780-4004 (inside NJ)

MANX[®]
software systems

WARE PUBLISHING • FRANKLIN • TELEVIDEO SYSTEMS INC. • PERKIN ELMER • NCR • SIERRA-O-LINE • NORTON CO. • HEWLETT-PACKARD • U.S. ROBOTICS • IRW • G.I.E. • SCOT • FLORIDA COMPUTER GRAPHICS • WESTERN UNION • MERRILL LYNCH • LOTUS DEVELOPMENT CORP. • MONOLITHIC SYSTEMS CORP. • HUGHES AIRCRAFT • ASHTON-TATE

system. If you press this key while entering text, **vi** will move the cursor to the beginning of the line you are working on.

SETTING UP YOUR TERMINAL

The **vi** set command allows you to set up your terminal to satisfy your personal habits and accommodate the type of work you are doing. This section describes the **wrapmargin**, **numbers**, and **wrapscan** set commands.

When you give a set command while in **vi**, it must be preceded by a colon and the word set (**:set**). You can also put set commands in the **.login** (C Shell) or **.profile** (Bourne Shell) file in your home directory

*As I think, I write, allowing my
whimsy to dictate to my fingers.*

so that UNIX executes them automatically each time you login. You can put as many set commands following the word "set" as you like. A typical entry in a **.login** file is:

```
setenv EXINIT 'set wrapmargin = 10 nowrapscan'
```

Under the Bourne shell, you can use the following commands in your **.profile** file:

```
EXINIT = 'set wrapmargin = 10 nowrapscan'
export EXINIT
```

You can get a list of all set commands and their current values by giving the command **:set all** while you are using **vi**.

Wrapmargin. The **wrapmargin** command specifies a right margin for your work in **vi**. When you set **wrapmargin**, **vi** will automatically insert RETURNS to move the cursor to the beginning of the next line whenever you enter text that reaches or passes the point you specified as your margin. This **vi** margin has nothing to do with the margins you specify for **nroff**.

Set the **wrapmargin** to the number of spaces you want between the right end of the longest line and the right side of your terminal. For 80-column terminals, this number is 80 minus the desired line length. To turn the feature off, set the **wrapmargin** to 0. Following are some sample commands:

```
:set wrapmargin = 10
:set wrapmargin = 15
:set wrapmargin = 0
```

Numbers. The **numbers** command specifies that line numbers are to appear while you are using **vi**. The line numbers are not part of the file and only serve to let you know what part of the file you are editing. Set **nonumbers** to turn this feature off. Following are sample commands:

```
:set numbers
:set nonumbers
```

Nowrapscan. The **nowrapscan** command specifies that **vi** is not to search past the end of the file when you use **/** to search for a word or regular expression. Similarly, this command causes **vi** to stop at the beginning of the file when you use **?** for backward searches. It also affects searches made with **n** or **N**. Set **wrapscan** to turn this feature off and allow searches to wrap around. Following are sample commands:

```
:set nowrapscan
:set wrapscan
```

List. As discussed earlier in this article, the **list** command specifies that **vi** display tabs as **^I** and mark the end of lines with a **\$**. Set **nolist** to turn this feature off.

TRICKS OF THE TRADE

Control Characters. Inserting or replacing control characters from **vi** can be a problem because **vi** interprets many control characters as commands. To insert a control character (such as CTRL-H or ESC) in a file, precede the character with CTRL-V. While in insert mode, press CTRL-V and **vi** will display a caret (^). Then press the character corresponding to the control character you want (press **H** for CTRL-H or ESC for ESC) and **vi** will display it. The **vi** editor displays control characters as the character preceded by a caret: CTRL-H appears as **^H** and ESC as **^[**.

You can use this technique within substitute and replacement strings (**:s** commands) to add or remove control characters from a file.

Executing Programs From vi. You can execute other programs while using **vi**. A typical use of this feature is running **spell** or **nroff** on the file you are editing. After you see any spelling or formatting errors, you can correct them immediately because you are still using **vi** to edit the file.

First, give the command **:w** to write out the file you are editing. This will allow **nroff** or **spell** to work on the current version of your file. Then, enter **!command** substituting the command you want to execute for the word "command." The **vi** editor will create a subshell to execute the command and, when the command finishes, return you to **vi**. The follow-

OUR 5620 TERMINAL WILL CHANGE THE WAY YOU VIEW YOUR UNIX* SYSTEM.

If you want to get the most out of your UNIX System, get the single terminal that performs like six—the 5620 Dot-Mapped Display from Teletype Corporation. This terminal is an exceptional value because it offers graphics and capabilities you'd expect to find only in higher-priced workstations.

With its unique capability to divide the display into six windows, the 5620 makes excellent use of UNIX System V resources to greatly improve productivity. You create and control the size and position of each window simply, using the electronic "Mouse." Unmodified host programs then view the windows as separate terminals, making it possible for you to work on several things at once.

Just think how much easier that'd make it for you to prepare documents and graphics, do computer aided engineering or write and debug programs. Imagine, for

example, while a program is compiling in one window, you edit the source code in a second window, check output in a third window, and send and receive mail concurrently in a fourth window.

The 15-inch diagonal display boasts 100 dots per inch, which gives you high resolution graphics and font capabilities. Complete with a full 32-bit processor and up to one megabyte of memory, you can also offload the host by running programs in the 5620.

As good as the 5620 makes Teletype Corporation look, it can make you look even better. To find out how, write Teletype Corporation, 5555 Touhy Ave., Dept. 3223-00, Skokie, IL 60077. Or call 1 800 323-1229, ext. 615.

TELETYPE®: VALUE SETS US APART.

"Teletype" is a registered trademark and service mark of Teletype Corporation.

*UNIX is a trademark of Bell Telephone Laboratories, Inc.



AT&T
Teletype Corporation

ing sample commands assume you are editing a file named "letter:"

```
:!spell letter
:!nroff letter | more
:!nroff -ms letter | lpr &
:!wc letter
```

After you give the command, you will see the output the command generates followed by the message "Hit return to continue." Press RETURN when you are finished viewing the output and are ready to return to **vi**.

This feature is also useful if someone writes to you (using **write**) while you are editing a file. You can respond, without leaving **vi**, by giving the command **!write name**. Substitute the name of the person you want to communicate with in place of "name." (If you were inserting text when you got the message, don't forget to press ESC to return **vi** to the command mode before giving the **write** command.) When you press DEL to exit from **write**, you can resume editing where you left off.

Counting Words. The **wc** (word count) utility, shown in the example above, displays three numbers. The numbers provide a count of lines, words and characters in the text. You can also call it from the command line by following **wc** with the name of the file you want a report on.

Blank Lines in nroff. When you are using one of the **nroff** macro packages (**ms**, **mm**, **me** or **mx**) and attempt to leave blank lines at the top of a document, **nroff** ignores the blank lines and puts the text at the top of the page. The **nroff** formatter insists that you put some text on the page before it will skip lines, and it does not count blank lines or regular spaces as text.

The **nroff** formatter considers a backslash followed by a space to be text. Once you put these characters at the beginning of a file, **nroff** will skip as many lines as you specify. The following example uses a comment (preceded by ****) to emphasize and preserve the space:

```
\ "this line begins with a quoted space
.sp 15 \ "skip 15 lines
Dear George:
```

The salutation will appear on the 15th line of the page following any space for the header. If you follow the quoted space with a comment, the procedure discussed earlier in this article will not remove the quoted space when you remove other trailing spaces.

THE **more** UTILITY

The **more** utility was developed at UC Berkeley and is available on most systems that support the

C shell. It is used primarily to look at files, one screenful at a time. But it also has many other useful features. To use **more**, give the command:

```
% more filename
```

substituting the name of the file you want to view for "filename." The **more** utility will display the first screenful of the file followed by:

```
--More--(xx%)
```

in reverse video. The "xx%" is the percent of the file you have already viewed. (The **TERM** variable for your terminal must be set to an appropriate **termcap** entry for the reverse video to work; if **vi** works properly, this variable has already been set.)

When you are finished viewing a screenful of text, press the space bar to display another screenful or press RETURN to display another line. Press DEL when you want to return to the shell.

Searching with more. You can use a slash in **more** to search for strings or regular expressions just as you do in **vi**. Instead of pressing RETURN or the

You can learn to use complex programs such as **vi** and **nroff** on a variety of levels.

space bar when **more** displays its prompt, press / followed by the string of characters (or regular expression) you want to find. Then press RETURN. You will see the words "...skipping" before **more** displays the next occurrence of the string near the top of the screen. To see the next occurrence of the same string, just press **n**.

Editing with more. If you decide you need to edit the file you are viewing with **more**, press **v** in response to the **more** prompt. The **more** utility will call **vi** with the number of the line you are viewing. When **vi** comes up, the cursor will be in the same area of the file you were viewing with **more**. When you exit from **vi**, you will go back to using **more**.

You can get a complete list of **more** commands by pressing **h** (for help) in response to the **more** prompt.

CHECKING YOUR WORK

Checking Macros. Sometimes, especially in longer documents, macros get mistyped or lines

that begin with periods get overlooked. The following one-line shell script can help detect this sort of problem by displaying a list of all the macros (all lines beginning with periods) in one or more files. The list will be in alphabetical order with only one listing per macro, no matter how many times it is used in the file. The macros are listed without arguments (.sp, .sp 2, and .sp 4 are all listed as .sp, for example):

```
grep '^\. ' $* | awk '{ print $1 }' | sort -u | more
```

You can leave the final "| more" off if your system doesn't have the **more** utility. After you put the above line in a file, give yourself execute access to the file with the following command (the command assumes that the file is named "printmacros"):

```
chmod u+x printmacros
```

Then use the shell script to check files by giving the following command:

```
printmacros file1 file2 file3 ...
```

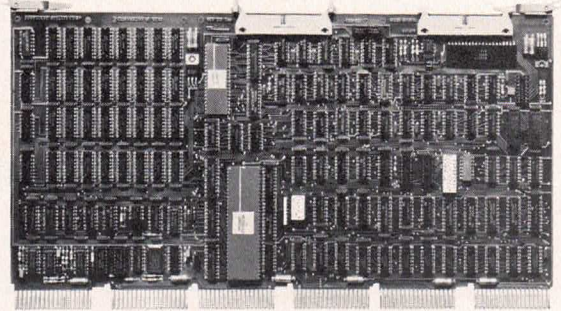
The shell script works as follows: the **grep** utility searches for all lines beginning with periods in all the files you gave as arguments to the command (\$*). (In the regular expression, the caret (^) indicates the beginning of a line, while the backslash causes the period to match only a period. Without the backslash, the period would match any character.) The output from **grep** is piped to **awk**, which displays the first word on each line (print \$1). The output from **awk** is piped to **sort** which is called with the -u option so that it displays a sorted list of only unique values (it only lists each macro once). The output is passed through **more** so that output larger than a screenful won't fly off the top of the screen.

Looking at Headings. Although some macro packages, such as **mm**, will automatically produce a table of contents, it is frequently useful to be able to obtain a list of heads for a document without running **nroff**. There are two basic styles of **nroff** macro commands that produce heads: those that precede the heading text on a separate line (the **.SH** and **.NH** commands of the **ms** macro package, for example) and the sort that take the heading text as an argument on the same line as the command (the **.H** command of the **mm** package is an example of this).

Listing ms Heads. The following **sed** script will

Continued on Page 89

DEC → TO IBM/SNA



Full SNA capability for your DEC computer! Comboard™/SNA gives your terminals access to IBM interactive applications. Data can be transferred between systems, all in the complete *fully supported package*. Comboard/SNA from Software Results.

Proven and reliable. Comboard/SNA is a single-board 256kb communications computer that plugs into your DEC Unibus. Teamed with Comboard software the system is a *cost-effective* solution to troublesome SNA communications problems.

Your DEC emulates an IBM PU Type 2 communication node. You have a *full gateway into your SNA* without passing through a secondary network.

For further information call or write Software Results... the leader in DEC to IBM communications.

COMBOARD™

Communications Results from

**SOFTWARE
RESULTS
CORPORATION**

Call Toll-free

1-800-SRC-DATA

(1-800-772-3282)

In Ohio call collect, 1-614-267-2203

2887 Silver Drive Columbus, Ohio 43211 Telex: 467-495 SRC DATA CI

COMBOARD is a Trademark of Software Results Corporation

DEC UNIBUS is a Trademark of Digital Equipment Corp.

Circle No. 285 on Inquiry Card

THE TROUBLE WITH TROFF

*The awful truth about
UNIX typesetting programs*

by Sandy Emerson



In a previous *UNIX REVIEW* article (October-November, 1983) describing **troff** formatting and typesetting, I may have been guilty of boosterism. The gist of that article was that do-it-yourself typesetting with a friendly local UNIX system was a possible and practical alternative to having documents typeset out of house. What I neglected to mention was the problems UNIX users are likely to experience when they try to put my advice into action. As a counterbalance, this article will discuss the darker side of the UNIX typesetting experience. Although there are still valid reasons why

UNIX users should learn to know and love the UNIX formatting and typesetting facilities – such as savings in costs and gains in convenience and control – considerable obstacles to the wider use of **troff** remain.

Informal surveys conducted by my colleagues and I indicate that the **troff** typesetting programs are under-used in both the end-user and commercial user communities. Of course, the **troff** programs in their current form could not be justifiably described as an end-user product. They are too hard to use, and the output devices necessary to produce high-quality typeset copy are simply too expensive.

The inertia of investment in non-UNIX typesetting technologies by the commercial typesetting industry also hampers the wider use of **troff**. Even though authors are submitting manuscripts these days in machine-readable form, publishers and typesetters alike seem to be wary of typesetting with a direct-from-disk method such as **troff**.

The practical problems that **troff** presents may be reinforcing the apparent conservatism of the publishing industry. These problems include difficulties with the **troff** programs themselves, concerns about the output devices, and headaches over the interaction between the two.

The awful truth about **troff** is that it is currently unsatisfactory in many ways, and that improvements to it are made only slowly and painfully. Perhaps the clearest proof of **troff**'s deficiencies is that UNIX-based typesetting services are now scarcely larger (higher volume) or more numerous than they were a year ago.

There are a couple of good reasons for revealing these facts about the bleakness of the UNIX typesetting scene. Perhaps this discussion will serve as both a warning to potential consumers to not bite off more than they can chew, and as a swift prod in the bit bucket for UNIX applications developers and typesetting equipment manufacturers.

The trouble with **troff** stems from two major sources: the programs themselves (as well as their fellow-traveler macro packages), and the output devices they drive. Further confusion arises from the way that the UNIX typesetting programs are being marketed.

Despite nearly 10 years of elapsed time (and many times more user-years of experience), the **troff** programs have not been made user-friendly by either direct reprogramming or the development of a handy front-end utility. The delay in performing major surgery on the code may be attributed to the fact that **troff** is quite large (7000 lines) and hard to understand. UNIX wizards who have been responsible for maintaining **troff** report that it is difficult enough to patch little leaks

**The awful truth about
troff is that it is
currently
unsatisfactory in
many ways, and that
improvements to it
are made only slowly
and painfully.**

that arise from time to time; revising the package would be too much to even contemplate.

Unfortunately, the author of **troff** is not available for consultation on a rewrite. As Bill Tuthill relates in "Typesetting on the Unix System" (BYTE, October 1983), "The **troff** program was written in PDP assembly language in 1973 by Joseph Ossanna. ... Updated programs were rewritten in the C language around 1975 and evolved slowly but steadily until late 1977, when Ossanna was killed in an automobile accident. Because nobody else knew

exactly how **troff** worked, its evolution came to a halt."

The **troff** programs thus are more or less like a black box, with facilities that can be used but not easily modified. The **troff** facilities are used by macro packages that provide a simplified interface to **troff** by defining and naming a set of standard formatting requests. The code is also called by pre-processor programs, such as the **tbl** program for typesetting tables, and post-processor programs, such as **col** for processing output destined for non-backspacing printing devices.

Tuthill does assert that, despite frequent and vociferous criticism of its design and coding, the **troff** source code has proved to be well-organized and extremely robust. But the fact remains that the code nonetheless seems to be too unwieldy and squirrely to modify.

In fact, when the illustrious Brian Kernighan took on the task of developing a typesetter-independent **troff** (making it compatible with a wider variety of typesetters) he used post-processor programs to avoid doing a major revision. As Kernighan points out in "A Typesetter-Independent TROFF" (Bell Laboratories, Murray Hill, New Jersey; Revised, March 1982), his transformation of the program consisted of causing it to output standard ASCII instead of "binary device codes specific to the (Graphics Systems) CAT (phototypesetter) and arcane beyond description." Several post-processing translation programs were made available with typesetter-independent **troff** to convert **troff** output into specific codes for specified typesetters. Kernighan also doubled the word size for the internal encoding of characters from 16 to 32 bits so that additional parameters could be added for use by the post-processors.

In his paper, Kernighan states that he refrained from revising **troff** from **main()** up because a) while he had thought about a better design, he was not yet satisfied with it; and b) a lot of

other software depends on **troff** in its current form – such as the macro packages, **eqn**, **tbl** and others. Kernighan also comments on **troff**'s "rebarbative syntax" and says that even Lorinda Cherry's C beautification program merely made the **troff** code legible, not comprehensible – lending support to our original contention that no one has revived or revised **troff** simply because it's too hard a job.

LEARNING TO THINK LIKE A TYPESETTER

In addition to the problem of inertia through bulk (where does a 1000-pound gorilla of a typesetting program sit? Anywhere it wants.), there is the problem that any serious user of **troff** must learn to think like a typesetter: either like the machine or like the printing professional. Henry McGilton, author of *Introducing the UNIX System* (McGraw-Hill, 1983), said in a recent conversation that the trouble with **troff** is that it is not an end-user utility but rather a language for micro-coding typesetting machines.

The only way to avoid getting too deeply into typesetting concepts and constructs is to use a macro package and accept all its faults and defaults. However, if the macro package's world view is not what you have in mind, you must immediately learn some typesetting terminology in order to fine-tune your text dimensions and page control in particular, and your font switches and point size changes in general.

So if you are content with the

ms macro package's default paragraph indent, you need never learn that the indent amount is 5 ens (the width of the letter "n" in the current point size), nor do you need to concern yourself with where the default value is stored or how it can be changed. But once you want to change anything in the way a macro package works or decide you want to use raw **troff**, you will need to grasp some basic typesetting terms – and some *programming* concepts to boot.

The programmability of the **troff** language is a strength, but learning about number and string registers, environments and quoting mechanisms is a bit of a trial for non-programmers. Manipulations of **troff** require a basic understanding of programming, and changes in register values (ems, ens, points, picas, *et al*) require some knowledge of typesetting.

Page control is the area in which **troff** is most noticeably perverse in requiring both typesetting and programming sophistication. Getting the program to give you a page break when the text fills to within some specified distance (such as one inch) from the bottom of the page is fairly straightforward – the macro packages do it for you automatically. But if you would like to do some special end-of-page processing, such as setting conditional page breaks depending on what comes next in the file, you will be thrown at once into a welter of typesetting and programming concepts.

For example, suppose you want top and bottom margins of approximately one inch, but you don't want the last line of a paragraph to appear by itself at the top of a new page, creating what typesetters know as a "widow." In order to prevent widows, you must either somehow squeeze that last line onto the same page as the rest of the paragraph or settle for a slightly larger bottom margin and divert some text onto the next page so that the widow line is no longer solitary.

If this typesetting were being done by hand, squeezing the widow line onto the previous page might be accomplished by slightly decrementing the vertical spacing between text elements (lines, paragraphs, section headings) on the page as a whole. However, it is this type of *ex post facto* manual repair for which computerized typesetting is not well suited. The more general-case solution (and the more easily programmable one) is to divert extra text to the following page to accompany the widow line.

Although this solution makes efficient use of **troff**'s programmability, it violates a stylistic goal of fancy typesetting – that of having a consistent bottom margin on every page.

The task of preventing widows must be undertaken with raw **troff** rather than with one of the standard macro packages. The **ms** package, for example, has literally dozens of lines of code controlling end-of-page processing, such as for page footers and footnotes. The definitions having to do with end-of page processing in the **ms** package have so many clauses and conditionals that revisions to suit a special case would be very difficult. The **ms** registers are filled and emptied, created and removed through some rather elaborate twists. Macros are even renamed in mid-stream. For example, the **FO** footer macro begins with the following lines:

```
.de FO
.rn FO FZ
```

The first thing this definition does is rename the macro from

YOU can print your **TROFF** manuals, proposals, forms...

right in your own shop!

Use **your** computer and our software (**XROFF**) and fonts (we have 100's) with the laser printer, typesetter, inkjet or dot matrix printer of your choice. We can provide whatever equipment you don't already have, at a price less than you would think.

Call or write for more information:

Image Network

770 Mahogany Lane, Sunnyvale CA 94806 (408) 746-3754

This ad was set using XROFF on a Xerox 2700 laser printer

UNIXTM & 'C'

HANDS-ON TRAINING

SEMINARS • VIDEO-BASED TRAINING • AND NOW INTERACTIVE VIDEO



Select your training medium according to the type of training you prefer and the number of people to be trained.

VIDEO-BASED TRAINING. The Computer Technology Group's Video-Based Training integrates professionally developed and produced video and text material, as well as hands-on exercises, into complete training programs.

Our courses are produced with the highest standards of video quality, applying the latest techniques of instructional design including the use of computer graphics and animation techniques to compress learning time. The students' time is not wasted with the "camera at the back of the classroom" or "chalk talk" approach which is so inefficient, and often ineffective, in transferring skills.

Our Video-Based Training courses are completely self-contained, including the hardware-independent hands-on exercises. All you need is a video cassette player.



COURSE

Computers at Work
UNIX—An Executive Perspective
UNIX Overview
UNIX Fundamentals
'C' Language Programming

Number of Modules	
Video-Based	Interactive

15

1

6

6

15

15

16

16

INTERACTIVE VIDEODISC TRAINING. Our new UNIX Videodisc Training Curriculum combines the benefits of our Video-Based Training with the flexibility of microcomputer access. Designed as a one-on-one tutor, our interactive system assesses the training needs of each student and dynamically tailors the training to his/her specific needs, thus eliminating redundant training. Through engaging exercises and interactive video, we are able to increase student comprehension while reducing training time.

Developed by the Computer Technology Group and Interactive Training Systems, the curriculum uses the latest laser videodisc technology—including IBM PC, color monitor and Interactive Training System Controller.

PUBLIC AND IN-HOUSE SEMINARS. Both public and in-house seminars are offered on a wide variety of UNIX and 'C' Language subjects, including:

UNIX Overview • UNIX Fundamentals for Non-Programmers • UNIX Fundamentals for Programmers • Shell as a Command Language • 'C' Language Programming • Shell Programming • Using Advanced UNIX Commands • UNIX Internals • UNIX Administration • Advanced 'C' Programming Workshop • Advanced 'C' Programming Under UNIX • Berkeley UNIX Fundamentals and "csh" Shell.

Call toll-free: (800) 323-UNIX
or in IL (312) 987-4082
310 S. Michigan Ave., Chicago, IL 60604

**COMPUTER
TECHNOLOGY
GROUP**
Telemedia, Inc.

"FO" to "FZ" in order to prevent a hypothetical new footer from interfering with the correct processing of the current one. If you try to work through such twists and turns to perform extensive revisions on a macro package, it may seem more as if you were doing macrame than macro making.

Another trouble with the **troff** programs and macro packages is their inconsistency. Take the units of measure for default values as an example. Default values come in two basic units of measure, the **em** for horizontally oriented requests, such as for indentations and page offsets, and the **V** (vertical space) for vertically oriented requests, such as spacing between lines. This differs both from what users might state in "English" and what printers might specify in printerese.

Default assumptions are also inconsistent between **troff** and the macro packages. The **ms** macro package, for example, has a default line length of six inches while **troff** has a default line length of 6.5 inches. A further question arises here: with so many other details specified in typesetting nomenclature, why are these line length defaults stated in inches? Printers, the people for whom the typesetting is being prepared, don't happen to think in inches - they expect typeset copy to be dimensioned in *picas*.

Inconsistency is further revealed as you gain enough expertise to make **troff** roll over and do tricks. You can't necessarily ex-

trapolate from one solution to a formatting or typesetting problem in order to solve another. Each solution usually turns out to be a special case, situation by situation, document by document.

The recital of perplexities and inconsistencies, and examples of the need to use and comprehend

To use a macro package is to accept all its faults and defaults.

both typesetting and programming concepts could go on and on, but I think you get the idea.

As if the **troff** program itself didn't give people enough trouble, there are still no inexpensive quality typesetters - or typesetter-like devices. There are three main problems with the output devices:

- they're too expensive
- they're too fragile
- program-processor interactions produce ugly output, for various reasons

First, all of the output devices that produce typeset with **troff** are too expensive for the average end user. The cheapest laser printer runs around \$3500. The

price of, say, an Autologic APS Micro-V is in excess of \$65,000. Which would you rather have, half a house or a typesetting machine?

Moreover, laser printer output is not good enough for production printing. Even if the fonts were pretty (which they're not, but this isn't solely the fault of the packagers), the output of a medium-priced laser typesetter is not suitable. The packagers of the various incarnations of the Canon LBP-10 laser printer reveal their awareness of this situation when they promote their **troff** and laser-printer systems as ideal for "internal documents" or "proofing typeset" or typesetting "rough drafts and in-house literature." (Who would ever consider typesetting rough drafts, anyway?)

Finally, there are program-product interactions that prevent the quality of the output from being as good as it could be. These include:

- font specification problems (the wrong slant for italics, wrong widths for special characters such as the = sign)
- poor resolution of output in the larger point sizes
- inability to run on many of the smaller UNIX systems without special installation or even custom porting

THE TROUBLE WITH MARKETING **troff**

Given that only selected UNIX installations may find it cost-effective to own a quality output device, the marketing of **troff** is something of a problem. The rest of the UNIX software as a whole is being promoted as "powerful and flexible," and perhaps even a little user-friendly (if **rogue** is included). Given **troff**'s unique user-hostility and the costliness of its toys, though, it's hard to decide how this product should best be "positioned."

Describing **troff** as a separate product is apt, even though it has heretofore come bundled with the rest of the UNIX system. AT&T Technologies and UNIX system

Image Network's XROFF, right now, prints *troff*/*ditroff* documents on:

- | | | |
|------------------------|----------------|---------------------|
| • VAX | • System 5 | • Xerox 2700 |
| • Pyramid | • System III | • Xerox 8700 |
| • Plexus | • Berkeley 4.2 | • Xerox 9700 |
| • PDP 11's | • IS/WB | • Diablo ink jet |
| • Integrated Solutions | • V 7 | • Diablo thermal |
| • Amdahl | • UTS | • Dec LNO1s |
| • IBM-PC | • MS/DOS | • Compugraphic 8400 |
| • 3B20 | • Xenix | • APS-5 typesetter |

at leading organizations from Berkeley to Murray Hill.

Call or write with your requirements!

Image Network, 770 Mahogany Lane, Sunnyvale CA 94086
(408)746-3754

This ad was set using **Xroff** on a Xerox 2700 laserprinter

Circle No. 80 on Inquiry Card

Announcing the WY-75.



Our new WY-75, VT-100* software-compatible terminal is specially fitted for use with your DEC* computer.

It offers a combination of features you can't find in any other VT-100 software-compatible terminal. Like a compact, ergonomic design. A finely sculpted, low-profile keyboard. And a swivel and tilt non-glare 14" screen, tailored with an 80/132 column format.

At a price of \$795, the WY-75 won't cost the shirt off your back.

Contact Wyse Technology for more information. And put your DEC on the best dressed list.

\$795

**VT-100 software
compatible and
dressed for success.**



WYSE

Make the Wyse Decision.

Wyse Technology, 3040 N. First Street, San Jose, CA, 95134, 408/946-3075,
TLX 910-338-2251, Outside CA call toll free, 800/421-1058, in So. CA 213/340-2013.

*VT-100 and DEC are trademarks of Digital Equipment Corporation.

packagers alike are, in fact, beginning to promote **troff** as a separate product instead of just another UNIX utility. Some UNIX packagers do not include **troff** with the rest of the system. Implementations of the Xenix system, for example, make the text formatting programs an optional purchase.

AT&T Technologies is now offering the spiffed-up version of

troff and other goodies as part of Documenters' Workbench, for UNIX System V.2. Interestingly, Documenter's Workbench includes a speedier version of the **nroff** formatter for ASCII terminals called **sroff** which is not compatible with either **troff** or regular **nroff**. Thus a value-added package solves one problem while creating several others. The **troff** family of programs will achieve a

position as a separate UNIX product, but what position is unclear.

WHITHER **troff**?

For the foregoing reasons and others not mentioned, **troff** is clearly not an end-user utility. Neither has a suitable commercial user interface been developed. However, there are some promising prospects on the horizon:

- The TYX Corporation of Reston, Virginia is marketing UNIX-based phototypesetting systems with a menu-driven, user-friendly front end. Although the TYX systems use the **TEX** typesetting program, the **troff** program is also available and may receive greater support from TYX in the future.

- The Golden, Colorado, branch of UniDot is now offering its soft-proofer (CRT and printer typeset previewer program) for **troff** as well as for **TEX**.

- The Bay Area's Image Network is offering the **xroff** program, an enhanced version of typesetter-independent **troff** that has been installed on about 35 different UNIX systems. The **xroff** program drives Xerox laser printers, the Diablo inkjet printer and the Compugraphic 8400 phototypesetter. A version of **xroff** also runs on the IBM PC, under MS-DOS.

- Technical Type and Composition of Salem, Oregon is now selling the Wizard typesetting package, which is an implementation of typesetter-independent **troff**. Wizard has a well-written user's manual that gives users real English words for the **troff** request names. Wizard is especially well-suited to driving Compugraphic typesetting machines.

Sandy Emerson is a freelance author living in Oakland, CA, who co-authored "The Business Guide to the UNIX System" and "Database for Your IBM PC" (both Addison-Wesley, 1984). She is currently at work on a book discussing UNIX typesetting.

/USR/TROFF™ ROUND TRIP NEW YORK/ PORTLAND \$36



From the Big Apple to UNIX™ Paradise and back. 5 minutes out, 48-hour return. TYPESETTING at 15½ feet-per-minute—wider than this entire UNIX™ Review page (about 1½" wider).

Telecommunication of documents, including mathematical formulae, gets TYPE SET and RETURNED on first class ACCUCOM-ODATIONS. No waiting at the TYPESETTING counter, and no reservations required. We also "book" passage for magnetic tape and raw text.

Call for details—1-800-ACCUCOM (222-8266)

UNIX is a registered trademark
of BELL LABORATORIES
Circle No. 46 on Inquiry Card



(503) 684-2850

9730 SW Cascade Blvd. / Suite 200 / Tigard, Oregon 97223

UNIX™ EVALUATION NEWS!!

AIM OFFERS APPLICATION - CONFIGURABLE UNIX BENCHMARK

DO YOU HAVE TO MAKE A UNIX PURCHASE DECISION IN THE NEAR FUTURE?

If you have to compare competitive UNIX systems in the marketplace *or* even evaluate the performance of your own system in light of new enhancements available, you know it can be a bewildering task.

THERE IS MORE TO COMPARING UNIX SYSTEMS THAN VENDOR STATISTICS ALONE.

How do you decide which UNIX system should be the standard for your company? You *could* compare published vendor data sheets but you would not be sure how they support your specific needs.

UNIX SYSTEMS RANK DIFFERENTLY WHEN COMPARED IN SPECIFIC ENVIRONMENTS!!

If your target needs are heavily database oriented, obviously the disk transfer rate of each system dominates your selection criteria, but if you need graphics capability, then the CPU resource becomes the center of evaluation. You might even need to select a system which does both functions well.

A UNIX BENCHMARK TAILORABLE FOR YOUR ENVIRONMENT IS AVAILABLE.

AIM Technology's second UNIX benchmark product, SUITE II, is parameterized so that you can evaluate how various UNIX systems, (such as Version 7, System III, System V, Xenix, etc.) would perform in your processing environment. Application characteristics (such as word processing, spreadsheet, graphics, communications, compilations and scientific requirements) can be weighted to reflect how much they are a factor in your overall evaluation. This testing can be run, tuned, and re-run on as many as 20 systems for comparison...

CALL OR WRITE FOR FREE "EVALUATING UNIX COMPUTERS" MANUAL*

FOR FREE MANUAL
In California Call
800-672-3470 x 815
All Other Areas
800-538-8157 x 815

This manual discusses how to evaluate the performance of various UNIX systems in specific processing environments. It will be most valuable to the multiple-purchase buyer who must decide which UNIX system *is* best suited for the application mix representative of *his* company's requirements.



AIM TECHNOLOGY

While the 60-day introductory 25% discount is in effect, AIM's Suite II Benchmark is available for \$2575. It can be ordered for immediate delivery from AIM Technology, 4655 Old Ironsides Drive, Suite 390, Santa Clara, CA 95050. For manual or additional information contact Ms. Jamie Mendez (408-727-3711).

™ UNIX is a trademark of AT&T Bell Laboratories

*Additional copies of the manual can be obtained for \$9.75

STRAIGHT TALK ABOUT MACROS

How to best tame the unstructured text

by Bill Tuthill

Unlike most text formatting programs, **troff** does not supply top and bottom margins, nor does it number pages. These functions must be performed by macros that you write yourself, or better yet, by canned macro packages. This article compares the four most widely-circulated macro packages available under UNIX.

When creating documents using UNIX, you're basically stuck with **troff** because existing documentation uses it and alternatives may not be available to your intended audience (who should, of course, be supplied with machine-readable text). So, how to make the best of it? You'll need a macro package that can do more than the critical function of pagination. It should also provide for different styles of paragraphs, displays, chapter titles, section headings, running page titles, footnotes and multicolumn output.

But, before going any further, just what is a macro? Simply put, it is something small that stands for something larger. Before **troff** begins to format text, it reads in macro definitions and stores them in a convenient place, such as a temporary file. Then, each time the small object (you defined) is encountered, **troff** can interpolate the larger object it represents. For example, a paragraph macro (often represented as ".PP") might

stand for the series of commands shown in Figure 1. It would be inconvenient to give all these commands every time you wanted to make a new paragraph. So, why not use the macro **.PP** instead? There is also a **.LP** (left paragraph) macro, which is much the same, except that it doesn't indent the first line. Some macro packages name these same two macros **.pp** and **.lp**, while others use only **.P** to signify paragraphs.

CRITERIA FOR COMPARISON

You have to decide what you need from a macro package. Do you want loads of features? Or do you want speed? The two are often mutually exclusive. If you pay for CPU time out of your own pocket, you would probably be willing to settle for a macro package that does little more than number pages. Unfortunately, there isn't one. Even the **me** macro package, which is the most economical of the ones available – and provides much more than page numbering – takes as much time being read in as it does formatting a single page of text. That's too much wasted CPU time if you're paying hard cash. On the other hand, if you own your computer and produce long complicated documents, you'll probably want a macro package that incorporates all the features, including multiple

columns, numbered footnotes, automatic table of contents and indexing.

On some computers, macro packages can be pre-interpreted so as to reduce CPU overhead whenever formatting begins. AT&T's System III and V have a facility for compacting macros so that output can start very quickly. The macro package is interpreted once beforehand and the system administrator saves the resulting **troff** temporary file in a directory named **/usr/lib/macros**. Each time a user specifies a compacted macro package, the saved file is quickly copied to the same place where the interpreted temporary file would otherwise go. If you are using System III or V, you will certainly want a macro package that supports this type of compaction.

Berkeley's 4.1 and 4.2 BSD, by contrast, keep interpreted macros in main memory, rather than in a temporary file. This arrangement greatly reduces I/O overhead. For one or two-page documents, compacted macros are faster than in-core macros, but for longer documents, it's much more efficient to keep macros in memory. So if you run Berkeley UNIX with a sufficient amount of main memory, you need not worry about support for compacted macros. Instead, you will want a macro package that does the conditional sourcing of macros

needed for special applications. Macro packages that attempt conditional sourcing do not even read the definitions of certain macros unless they are used in your text.

System V Release 2 is reputed to have combined both approaches. Compacted macros are available and a compile-time option for keeping interpreted macros in memory is included, assuming your computer has enough main memory. When you employ compacted macros with in-core macros, the saved file is copied into memory rather than into a temporary file. In this case, you would probably want to use a macro package offering both compacted macros and conditional sourcing. This would allow formatting to start up quickly, and would also use as little memory as possible for each job.

There are four general-purpose macro packages available to most UNIX users: **ms**, **me**, **mm** and **mx**. The first three are quite different, but the fourth is similar to and, in fact, was derived from **ms**. Your computer may not have all four packages, but there's a good chance you could obtain any of the ones that are missing. The **mm** macros require a System III or V license. The remaining three require only a Version 7 license, which is included under a System III or V agreement.

THE BIG FOUR

The **ms** (manuscript) macros were written by Mike Lesk, with help from others at Bell Laboratories Research in Murray Hill, NJ. Lesk's work provided one of the first **troff** macro sets, so it could not benefit from hindsight, as other macro packages have. Nonetheless, the **ms** package has proved remarkably useful and long-lived. Although it supports several internal Bell Labs document formats, it can be used for applications of all sorts, because the macros can be easily modified and extended.

The **me** macros were written by Eric Allman, while he was an EECS student at UC Berkeley.

They display much cleverness, but because of their obstinate refusal to duplicate upper case names in **ms**, many of internal registers have inscrutable names. On the whole, they are more bug-free than **ms**, and are more easily extensible as well. Their main strength shows in the production of Berkeley theses. But for general typesetting, **me** makes it difficult to control leading (i.e. vertical spacing) separately from point size.

The **mm** (memorandum) macros were written by D.W. Smith and John Mashey, who were part of the group that developed Programmer's Workbench. Although the **mm** macros are full of wonderful features, they are bureaucratic in orientation and may be too slow and complicated for general typesetting applications. They are the most bug-free of all the packages and supply good error messages when the user does something wrong. One nice feature is that many number registers can be controlled from the command line without making any changes in text. Unfortunately, **mm** provides so many options that it creates an extra level of confusion that macro packages should strive to avoid.

The **mx** (experimental) macros are a revised version of **ms** written at the UC Berkeley Computer Center that combine many new features from **me** and **mm**. The designation indicates that, like the missile system of the same name, the macros may be outmoded five years after they were built. The **mx** macros replaced **ms** on 4.2 BSD, where the old **ms** is now called **mos**. Since **mx** was intended to be upwardly compatible with **ms**, there are inherited design flaws and some of the code is incomprehensible. Although quite a few bugs were fixed, floating keeps still don't work properly in all cases and the error messages are not as good as those from **mm**.

Here are some timing statistics for the four packages. All benchmarks were obtained on a quiescent Z8000-based system

that yielded strikingly similar results no matter how often the tests were run:

```
% nroff -ms null
9.9u 3.3s 0:18 73%
% nroff -me null
4.2u 1.5s 0:09 41%
% nroff -mm null
17.6u 4.0s 0:25 86%
% nroff -mx null
4.9u 1.0s 0:09 42%
```

Compacted macros were not used. As stated above, these timing statistics are relevant only if you can't use compacted macros.

The following list provides the number of characters in each macro package. The size of a macro set determines how long it takes to read and is an indication of the number of macros defined:

-ms	31,637 bytes
-me	21,310 bytes
-mm	76,147 bytes
-mx	25,492 bytes

The small size of **me** results from stripping its comments out. With comments, it would have been 32,276 characters long. The **mm** macros have also had their comments stripped, but the commented version is unavailable outside of AT&T, so its size cannot be recorded here. Figure 2 compares the four macro packages according to certain distinguishing features. As you can see, the **ms** macros have fewer features than any of the other three. However, each package is missing something. When choosing a macro package, you must decide which features you need most and which ones you can live without.

PUBLICATION FEATURES

The first group of features listed in Figure 2 are those that would be useful in publication environments for producing books and full-length manuals. Multiple columns make text easier to read than single columns do because the eye can move from one line to the next without getting lost. All four macro packages provide for multicolumn text. Generally you

```

.de PP          define PP macro
.ps \\nP       reset point size to normal
.vs \\nV       reset vertical spacing to normal
.ft 1          reset to Roman font
.ll \\nLu      reset line length to normal
.ne 2.1v       make sure there's enough space to avoid widows
.sp .5v        put in a half-line space
.ti +5n        indent the first line
..            end macro definition

```

Figure 1 — The .PP macro.

get two columns, but by specifying column width, you can get three and more columns from any of the packages.

When books and manuals are printed on both sides of the paper, you want page numbers to appear on the outside of the document. This means that they should fall on the right side of odd-numbered pages, but on the left side of even-numbered pages. The **ms** macro package is the only one that does not provide for different headers and footers on even and odd pages. None of the four macro packages, however, can produce multi-line headers or footers, which are required by some professional conventions.

One of the features that the **scribe** formatting system provides, but **troff** has been slow to match, is the automatic generation of tables of contents. The **ms** macros provide nothing for this. The **me** macro package is a little more modern, but still does not automatically collect section headers. You need to specify each table of contents entry separate from the section header, as in the following:

```

.sh 1 "The Importance of Being Earnest"
.(x
The Importance of Being Earnest
.)x

```

The **mx** macros have table of contents support similar to this and also provide a means of producing a table of contents as an

afterthought. The **mm** macros are the only ones that give you what **scribe** does — that is, when you specify a header, **mm** automatically collects it for inclusion in the table of contents. The following is a sample of **mm** header input:

```
.H 1 "The Importance of Being Earnest"
```

That's all you need to say, until the end of your document, when you can call up the table of contents by using the **.TC** macro.

Neither **ms** nor **mm** provide facilities for producing book indexes. The table of contents apparatus in **me** could be adapted to do this, but it would require some fiddling. The **mx** package provides a simple **.IX** macro, defined as follows:

```

.      \" IX - index words to stderr
.de IX
.tm \\$1\\t\\$2\\t\\$3\\t\\$4 ... \\n(PN
..

```

The user specifies words to be indexed, up to four levels deep. External software then takes care of details such as alphabetizing and formatting the index. The above macro definition of **.IX** could be expropriated for use in any of the other three macro packages.

ACADEMIC FEATURES

The second group of features noted in Figure 2 are those that would be useful for producing papers and books in an academic environment. Accent marks are necessary for printing text in

foreign languages. The **ms** macros provide all the diacritics necessary for French and German, and also a haček (for Czech). The **me** macros have all these accents available, in much higher quality, in addition to the Å for Swedish. The **mm** macros provide everything **ms** does, except the haček, and the quality is better than in **ms**. The **mx** macros, though, are alone in providing the upside-down question mark ¿ and exclamation mark ¡ for Spanish, the ø for Norwegian, the underdot ̣ for South Asian languages, the macron ā for Latin, and all the necessary symbols for Old Norse and Middle English.

Academic papers generally contain many footnotes. Some even have more footnotes than text. While all four macro packages have facilities for footnoting, only **me**, **mm** and **mx** provide for numbered footnotes, which are crucial in the academic environment. The numbering of footnotes allows you to have many of them on the same page and also helps organize references. Note that the **refer** program is a useful tool for this task, but it is not supported by the **mm** macros. The **refer** program is a preprocessor for bibliographies, just as **tbl** is for tables, and **eqn** is for equations.

Many academic standards recommend endnotes instead of footnotes. Endnotes appear at the end of a paper, whereas footnotes appear at the bottom of the page where they are referenced. This particular academic standard was devised in the days before computers were widely used because of the prohibitive expense of setting footnotes. Even though computers now make this task cheap and easy, academic standards are slow to catch up. The **ms** and **mm** macros provide no facilities for endnotes, although in conjunction with **refer**, the **ms** macros can produce endnotes. The **me** and **mx** macros have

In microcomputers today, UniSoft sets the standard.

AT&T has recently been advertising that their UNIX™ operating system will be the standard OS for microcomputers. That's true. But if you want AT&T's UNIX software on micros today, talk to UniSoft Systems.

UniSoft has been delivering AT&T's UNIX adapted for 68000-based microcomputers for two years. More than 75 different computer systems run the UniSoft software, UniPlus+.™ At each Bell release level, all these systems are object code compatible. This means that applications software developed on any UniPlus+ system will work on any other. This is where software portability pays off.

UniSoft enhances Bell's vanilla UNIX with the best features from the Berkeley BSD research version of the UNIX operating system. IP/TCP networking, record and file locking, and virtual memory from UniSoft turn UNIX into a commercial product. All this added value is still Bell-compatible.

Don't wait six months to get System V running on your hardware. UniSoft's customers can ship it now.

If you're building or selling a 68000-based UNIX system, your operating system should come from UniSoft Systems, the UNIX experts.



THE BERKELEY PORT AUTHORITY

Circle No. 287 on Inquiry Card

*UNIX is a trademark of Bell Laboratories

739 Allston Way, Berkeley, CA 94710 • (415) 644-1230
TWX II 910 336-2145 • UUCP ucbvax!unisoft!unisoft

DOCUMENT PRODUCTION FEATURE

Macros Broken Down by Features				
<i>Nifty Features</i>	<i>-ms</i>	<i>-me</i>	<i>-mm</i>	<i>-mx</i>
Multiple Columns	yes	yes	yes	yes
Even/Odd Headers	no	yes	yes	yes
Table of Contents	no	yes	yes!	yes
Indexing Support	no	limited	no	yes
Good Accent Marks	some	many	some	most
Numbered Footnotes	no	yes	yes	yes
Option for Endnotes	no	yes	no	yes
Multi-Level Headers	no	no	yes	no
Numbered Paragraphs	no	yes	yes	no
Itemized Lists	use .IP	use .ip	yes	use .IP
Bell Labs Garbage	some	no	yes	no
Berkeley Garbage	no	yes	no	some
tbl and eqn Support	yes	yes	yes	yes
refer Support	yes	in 4.2	no	yes
pic and ditroff Support	no	no	no	no
Compacted Macros	no	no	yes	yes
Conditional Sourcing	no	yes	no	yes
Robustness	poor	fair	good	fair
Readability	fair	poor	poor	good
Comments Available	yes	extra	no	yes
Portability of Text	great	poor	fair	good

Figure 2

commands for producing endnotes, although the **me** method (delayed text) will overflow after the first few pages of notes.

BUREAUCRATIC FEATURES

The third group of features listed in Figure 2 are those that would be useful for document production inside a large company. All four macro packages provide numbered section headings that help give writers the illusion of good organization. The **mm** macros are alone, however, in providing multi-level headers that change in appearance from one level to the next. Main section headers are set in boldface, on a line of their own, while sub-section headers are set in italics, flush left with the text that follows. As discussed above, all section headers are automatically collected for the table of contents, which can be printed at the end of a document.

Automatic paragraph numbering is another device used to help schematize prose. Both **me** and **mm** provide numbered

paragraphs, while **ms** and **mx** do not. The lack of this feature is rather strange, since numbered paragraphs are trivial to implement and can be very useful at times.

One of the salient features of bureaucratic prose is the bullet paragraph. This device allows writers to demonstrate:

- a talent for simplification, and
- a penchant for snappy phraseology

Bullet paragraphs provide a way of generating an itemized list. By simply replacing the bullet (•) with another character, you can produce a different form of itemized list. Of the four macro packages discussed here, only the **mm** macros provide support for itemized lists. The others can do itemized lists in a somewhat roundabout way – by using indented paragraphs for each item on the list.

Two of these macro packages come from inside AT&T, the world's largest corporation, while

the other two originated at UC Berkeley, where the administration building, Sproul Hall, strongly resembles Kafka's Castle. As a consequence of intrusive bureaucracy, all four packages contain excess garbage from their respective institutions. For example, the **ms** macros still have **.EG** to produce Engineer's Notes, an internal Bell Labs format no longer in use today. The **me** macros, on the other hand, offer you **.th** for Berkeley's thesis mode, something not frequently used at other institutions.

It may seem that the various formats provided by the **mm** macros would not be useful elsewhere. But large corporations often find that the **mm** formats can be adapted to their own internal styles with a minimum of difficulty. So if this is important to your institution, you probably will want to use the **mm** macros. After all, the larger the corporation, the greater the need to write memoranda.

PREPROCESSOR SUPPORT

All four macro packages have special macros for **tbl** and **eqn** since these two preprocessors have been around for a long time. The **ms** package in Version 7 was the first to support **refer**, and the **mx** package has improved on this support. The **me** macros provided nothing for **refer** until 4.2 BSD was released, at which point they were extended to provide for bibliographies. The **mm** macros are alone in their refusal to support **refer**. This is partly the result of AT&T's commercial orientation, but another problem is that **mm** is already so big that it cannot grow much larger without straining the limits of **troff**. Something needs to be removed before **refer** support can be added.

Commonly available versions of these four macro packages

CommuniqueTM

The PC-to-UNIXTM Connection.

- Host control of PC program execution.
- Simple, elegant file transfer.
- DEC and IBM terminal emulation.
- Full TERMCAP compatability.
- UNIX utilities to control file transfer and program execution.

**From the company that put UNIX
on the IBM Series/1.**

Call Keith Burleigh at 313-665-8778.

COSI

313 North First Street

Ann Arbor, Michigan 48103



Communique is a trademark of COSI.
UNIX is a trademark of Bell Laboratories.

provide no support for device-independent **troff**, called **ditroff**. For one thing, more than four fonts are available with **ditroff** and there are also new graphic primitives for drawing lines, circles, ellipses, arcs and splines. These graphic primitives are used by the **pic** preprocessor for drawing diagrams. There is a version of **ms** that supports **pic** on the **ditroff** distribution tape, and there are certainly new versions of **mm** and **mx** somewhere that support these graphic primitives. Actually, only a few macros have to be added.

SPEED AND RELIABILITY

The last six categories in Figure 2 have to do with various programming considerations such as efficiency, robustness, maintenance and portability. Macro compaction and conditional sourcing are two techniques for achieving increased efficiency. The **ms** package offers neither, **me** offers conditional sourcing, **mm** offers macro compaction and **mx** offers both. Macros intended for conditional sourcing are never compacted, of course, and will have to be interpreted if and when they are sourced.

The **mm** macros are the most robust of the four packages. They continue to function in the face of input error and they give good error messages when something goes wrong. In most incarnations, they are also relatively bug-free. The **me** macros and the **mx** macros score a bit lower because they don't have great error messages and they aren't as well maintained as **mm**. The **ms** macros are the worst of the four from the standpoint of reliability. There are cases where **troff** can go into an infinite loop when you forget just one **ms** macro.

Most people soon find that they need to change their macro

package in some minor way. For example, you may want to devise a macro to print the name of your institution. When you get to this point, you will find that a macro package must be readable and should be supplied with comments that help you figure out what's going on. Unfortunately, the **mm** macros have no comments whatsoever, and contain spaces only where syntactically necessary. This makes them difficult to read and maintain. The

**As a consequence of
intrusive bureaucracy,
all four packages con-
tain excess garbage
from their respective
institutions.**

lack of spaces and comments might save a bit of CPU time, but only if you're not using compacted macros. If you use compacted macros, you save nothing - so in the long run, it hardly seems worthwhile to leave the comments out.

The source for the **me** package contains comments, but some people find the internal names of **me** macros and registers difficult to comprehend because they contain non-alphabetic characters. On the other hand, indentation and white space are used to good advantage in the **me** source code. The **ms** and **mx** macros are somewhat more readable than **me**, if you understand **troff**, and **mx** uses indentation in much the same way as the **me** macros do. Both **ms** and **mx** have

quite a few comments, though not as many as the commented version of **me**.

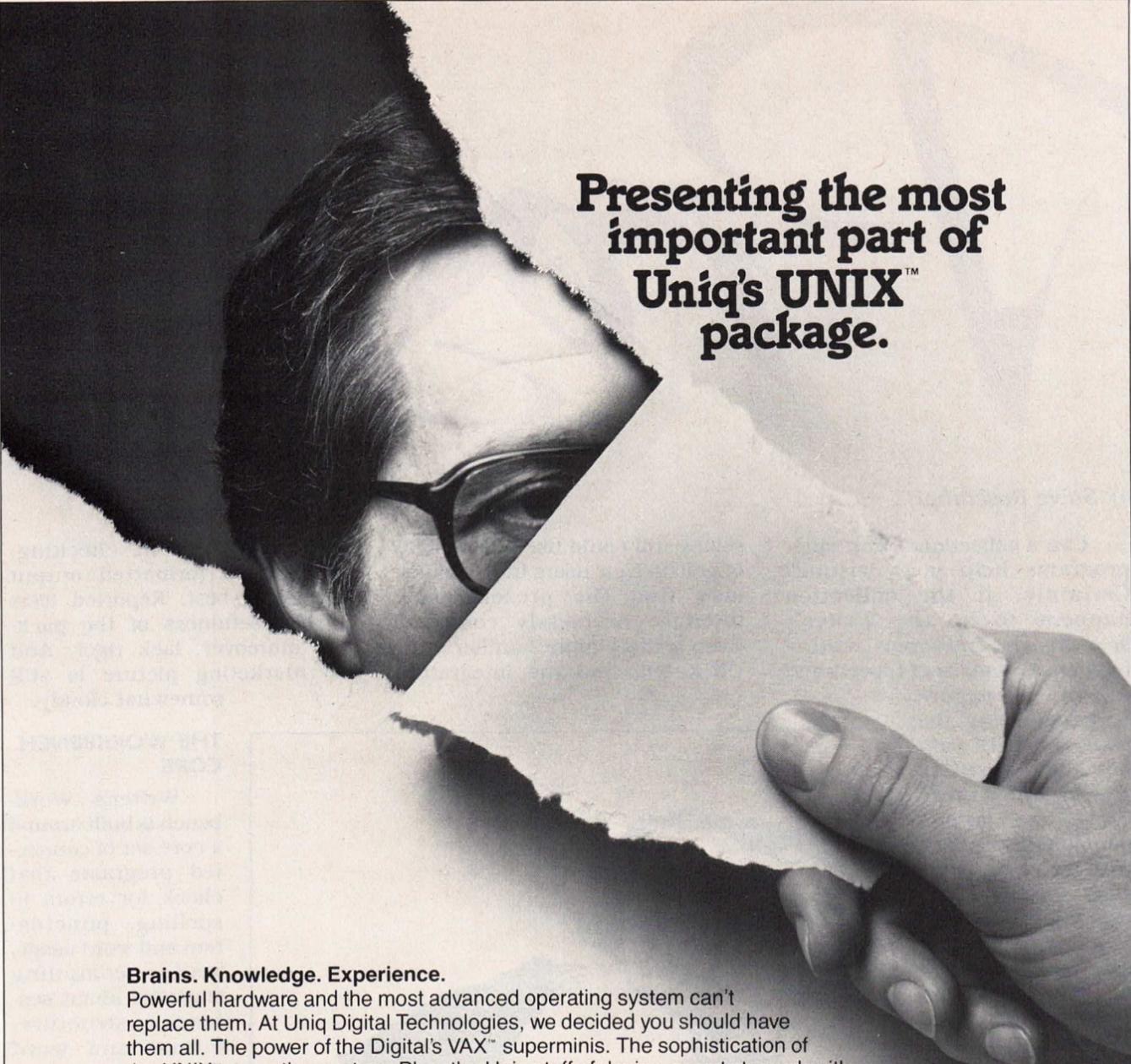
The final question to consider is, how portable will your text be if you decide to use any one of these four macro packages? Text formatted with **ms** commands is the most portable since every release of UNIX since Version 6 has contained **ms**. This is true because even though AT&T's System III and V dropped support for the **ms** macros, almost every vendor who has released a System III or V-based computer has also included **ms** for backward compatibility.

Text formatted with **mx** is not as portable, unless you stick with commands available in **ms**. For example, if your text contained commands for creating a table of contents and someone tried to format it on a system with the old **ms**, all your **mx** commands would be ignored.

Text formatted with **mm** is even less portable, although this situation will improve in the future as more and more systems become based on AT&T UNIX. At the moment, however, UNIX systems based on Version 7 may not have the **mm** macros available.

Text formatted with **me** is the least portable, because many UNIX systems don't have Berkeley enhancements. Even those that do include **me** only after accounting for **termcap**, **vi** and the C shell. If you send text formatted with **me** to an installation, you may wish to send the macro package along as well.

Bill Tuthill was a leading UNIX and C consultant at UC Berkeley for four years prior to becoming a systems software analyst at Imagen Corporation. He enjoys a solid reputation in the UNIX community earned as part of the Berkeley team that enhanced Version 7 (BSD 4.0, 4.1 and 4.2). ■



**Presenting the most
important part of
Uniq's UNIXTM
package.**

Brains. Knowledge. Experience.

Powerful hardware and the most advanced operating system can't replace them. At Uniq Digital Technologies, we decided you should have them all. The power of the Digital's VAX[™] superminis. The sophistication of the UNIX[™] operating system. Plus, the Uniq staff of design experts to work with you and bring it all together.

Behind every Uniq UNIX package is a complete support plan. Support that includes the broadest variety of UNIX flavors offered by any Vendor—including the latest versions from AT&T and Berkeley. Additional capabilities include custom driver development, kernel modification and training classes in basic UNIX user skills, system administration and advanced C language programming. We've established a Customer Technical Assistance Center staffed with experienced software engineers to provide solutions to those critical calls. And our on-line diagnostics system is designed to find the problems before the solutions get expensive.

Uniq also offers software tools, including UNIFY[™] relational data base, Unicalq electronic spreadsheet, compilers and other software development and office automation products.

Today you can't only depend on the computer environment to solve your company's problems. You need to look beyond them to the most important part of the package. Support. Uniq support.

For more information on how Uniq can make it all work for you, call or write for our free descriptive brochure.

UNIX is a trademark of AT&T Bell Laboratories
VAX is a trademark of Digital Equipment Corporation
UNIFY is a trademark of Unify Corporation
Copyright 1984, Uniq Digital Technologies



DIGITAL TECHNOLOGIES

28 South Water Street, Batavia, IL 60510 • (312) 879-1008

We make it work.

AUTHORIZED **digital**[®] COMPUTER DISTRIBUTOR

Prose w

An Overview

by Steve Rosenthal

Can a collection of computer programs help your writing? Certainly, if the collection happens to be the Writer's Workbench and your writing happens to consist of reports and technical papers prepared using the standard UNIX editors and formatting macros. The Writer's Workbench won't completely replace proofreading and copy editing, but it can spot a large proportion of whatever typographical and linguistic errors exist in the documents it processes.

Many UNIX users now routinely run anything they write through one or more of the workbench programs, if only to spot the most egregious errors. The package also is being used by several universities and a number of corporate training centers to help teach writing skills. Acting as a private tutor, the programs allow students to improve their writing style and produce better documents.

However, the programs them-

selves still could use a good deal of polish. New users in particular may find the present user interface needlessly complex. Even writers more familiar with UNIX will find the integration

between document checking, editing and formatted output indirect at best. Reported tests of the usefulness of the package, moreover, lack rigor. And the marketing picture is still somewhat cloudy.

THE WORKBENCH CORE

Writer's Workbench is built around a core set of connected programs that check for errors in spelling, punctuation and word usage, while compiling statistics about sentence structure, format and word classification. Figures developed by the programs are then compared to values considered to be indicative of good writing and the results are displayed and explained. These linked programs are supplemented with an extensive set of utilities, help programs and stand-alone tools.

You can run the full set of connected programs by invoking **wwb** or you can run many of the programs individually. For instance, to see the results of the



Illustrations by Erik Jorgensen

ith Style

of Writer's Workbench

proofreading programs alone, you could select **proofr**. Selecting **prose**, on the other hand, would yield an assessment of your document based on a range of statistics. The occasion may arise when you will want the system to check only one particular aspect of your document, so the option of running most of the subsidiary programs individually is welcome.

The **proofr** program concentrates on finding errors in individual words and phrases. To find spelling errors, it calls on **spellwbb**. For checking punctuation, it invokes **punct**. In practice, if you need much help from these two programs, you ought to define your own command or create a shell script to invoke just this pair since most of the other workbench programs can't do a very good job unless your spelling and punctuation are correct.

Words in your document are checked against a standard dictionary list and a supplementary dictionary

of your own devising by the **spellwbb** program. Any word not found in either dictionary – and not derivable from words that are by using standard prefixes and suffixes – is flagged as a possible

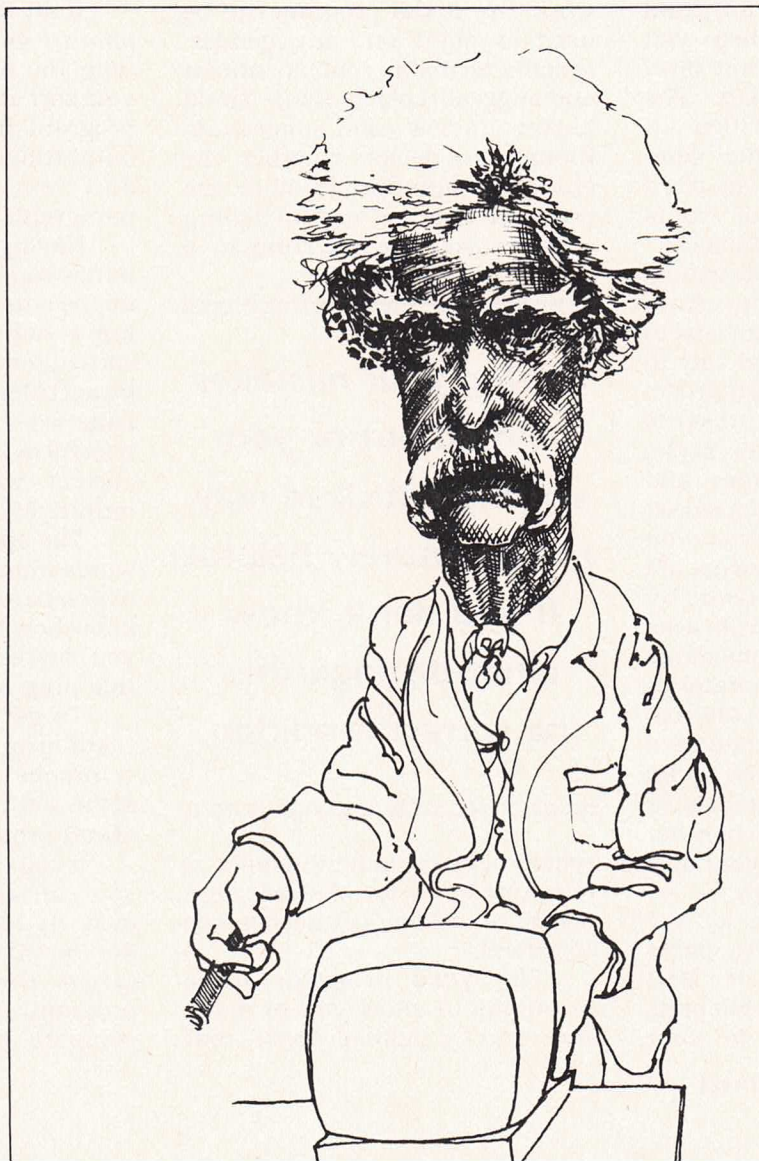
misspelling.

The **punct** program applies a set of rules for English-language punctuation to your input document, noting any errors it finds. Because most grammarians agree

on punctuation, this program will even correct punctuation in your file when run as a standalone program with the right options set.

The **double** program, another proofing utility called by **proofr**, checks your files for two or more uses of the same word – a condition that is very very rarely correct. Doubled words occur more frequently with electronic editing, especially when a line-oriented editor is used to change paragraph-oriented text.

The **proofr** portion of Writer's Workbench also invokes **diction**, a program that compares text to a list of phrases that writing authorities consulted by the authors of the workbench have cited as likely to be overused, incorrect or confusing. Output from **proofr** lists suspect words or phrases



along with some suggested replacements. Some of the listings are undoubtedly good advice – but in other cases, you'll probably feel that you like your own text better than any of the listed alternatives.

A last part of the **proofr** program checks your text for split infinitives (constructions such as "to quickly run"). Again, because there are instances where the split form best conveys a message, the opinions are presented as advisory warnings for you to accept or reject.

CHECKING YOUR STYLE

While **proofr** looks for errors that shouldn't be left in any kind of text, **style** tries to help you decide whether the flow and level of your text is appropriate. The program gathers statistics on sentence length, sentence complexity, reading grade level, word usage and the types of words you've used to start sentences.

Unless you're a statistical grammarian, the output from **style** probably won't mean a lot to you – so the more usual way to view the results is through **prose**. The **prose** program presents the results compiled by **style**, explains their significance and compares them with a selected set of normal values. Unless you specify otherwise, the norms are a set of values derived by averaging results taken from a set of technical papers considered well-written by Bell Laboratories department heads. As an alternative, you can specify an option to use an average derived from Laboratories training materials, or even an average that you or other users have derived from your own set of favorite documents.

The **style** program, by the way, is an outgrowth of the **parts** program developed at Bell Laboratories to identify what parts of speech each word in your

document represents. The **parts** program was originally written, developer Lorinda Cherry says, as part of a research project aimed at adding the correct intonation to a voice output system.

THE STANDALONE PROGRAMS

In practice, **wwb** invokes **proofr** and **prose**, which in turn invoke a second layer of programs. The workbench also includes close to a score of other programs that you can run independent of the **wwb** command.

So if you're writing for a publication and you feel social gender prejudice may have slipped unconsciously into your work, the **sexist** program can be used to point out any gender-specific terms in your document and suggest replacements. You'll have to review each suggestion, though, to decide whether the change is appropriate since the program has no way of telling whether you are referring to a

your audience might not already know well. This program is a bit on the simple-minded side (it merely looks for successive capital letters), so if you want to be especially careful, you might also run **spellwwb** to look for unfamiliar words.

The **findbe** program outputs your document with all the instances of the verb "to be" highlighted. According to Bell's grammarians, over-reliance on this verb should be avoided. In addition, it is one of the components of the passive form of expression, so its occurrences may well serve as flags of excessive passive phrasing.

If all the details of your text obscure your view of its organization, the **org** program may help you sort it out. This standalone program filters your document, outputting only headings and first and last sentences of each paragraph.

Having your mistakes pointed out or your writing criticized may not be entirely helpful if you don't know what distinguishes the correct rendition. So the programs **punctrules**, **worduse** and **splitrules** serve as online grammatical references discussing punctuation, correct word usage and split infinitives, respectively.

The **spelltell** program will list words from the dictionary list that match patterns you provide. If you know how to spell a part of a word, you can retrieve the rest of it using this program.

To get an explanation of the standards used by **prose** to comment on your document's **style** statistics, run the **prosestand** program.

If you don't like the standards, you can always substitute your own. By running a couple dozen documents representing your target through the **mkstand** program, you can have **prose** compare your documents to a

Having your mistakes
pointed out or your
writing criticized may
not be entirely helpful
if you don't know
what distinguishes
the correct rendition.

specific person (where gender is relevant) or to an abstract case (where sexual identity need not be specified).

The **acro** program spots acronyms to allow you to make sure you've defined those that

68000

16000

8086/88

Z8000

SOFTWARE DEVELOPMENT TOOLS

"One-Stop Shopping"

OASYS provides a "One-Stop Shopping" service for software developers and managers in need of proven, cost effective, **cross-** and **native-** development tools.

OASYS can save you time, energy and money! We understand what it means to be a developer. Over the past 3 years, we've built over 1MB of working code.

We not only develop our own tools, but also specialize in evaluating, selecting and distributing the best complementary tools from other suppliers.

Our tools are currently in use in over 1,000 installations worldwide on **micro-, mini-, and mainframe** computers for a variety of **8-, 16- and 32-bit** UNIX (and non-UNIX) systems.

Most likely, we have what you're looking for (even if it doesn't appear in the tables shown). But, if we don't, we'll be glad to tell you who does.

So, call or write today for more information and start shopping the smart way, the fast way, the economical way.

"The One-Stop Shopping Way."



OASYS

60 ABERDEEN AVENUE
CAMBRIDGE, MA 02138
(617) 491-4180

CROSS TOOLS

PRODUCTS ⁽¹⁾	HOST ⁽²⁾	TARGET ⁽³⁾
C COMPILERS	VAX, PRIME	68000 16000 8086/88
PASCAL COMPILERS	VAX PDP-11, LSI-11 PRIME	68000 16000 8086/88
FORTRAN COMPILERS	VAX PDP-11, LSI-11	68000 16000 8086/88
ASSEMBLERS ⁽⁴⁾	VAX, PDP-11, LSI-11, PRIME, IBM/PC, IBM 370	68000, 16000, 8086/88, Z8000, 680X, 808X, Z80
SIMULATORS	VAX, PDP-11 LSI-11, PRIME, IBM/PC, IBM 370	68000, 8086/88 808X, Z80

(1) WE DISTRIBUTE PRODUCTS FOR: GREEN HILLS SOFTWARE, VIRTUAL SYSTEMS, COMPLETE SOFTWARE, PACER SOFTWARE, SOFTWARE MANUFACTURERS

(2) HOST OPERATING SYSTEMS INCLUDE: VMS, RSX, RT-11, PRIMOS, UNIX V7, III, V, BSD 4.1, 4.2, UNOS, IDRIS, XENIX, MS/DOS, VM/CMS, CPM 68K

(3) OTHER TARGETS ARE: M6801-6803, 6805, 6809, 8080, 85, 28, 35, 48, 51; Z-80

(4) ALL ASSEMBLERS INCLUDE LINKER, LIBRARIAN AND CROSS-REFERENCE FACILITY

(5) AVAILABLE ON: CALLAN, OMNIBYTE, CHARLES RIVER DATA, PLEXUS, SAGE, FORTUNE, WICAT . . . to name a few.

C/UNIX NATIVE TOOLS

- NATIVE ASSEMBLERS FOR 68000s^(4, 5)
 - SYMBOLIC C SOURCE CODE DEBUGGER
 - C-TIME PERFORMANCE UTILITY
 - UP/DOWN LINE LOAD UTILITIES
 - COMMUNICATION UTILITIES
 - BASIC-TO-C TRANSLATOR
 - C-BASED FLOATING POINT MATH PACKAGE
- ... AND MORE

TRADEMARKS: UNIX IS A TRADEMARK OF BELL LABORATORIES, XENIX AND MS/DOS ARE MICROSOFT CORP'S, IBM/PC, VM/CMS, AND IBM 370 ARE INT'L BUSINESS MACHINES, VAX, PDP-11, LSI-11, VMS, RSX, AND RT-11 ARE TRADEMARKS OF DIGITAL EQUIPMENT CORP, CPM 68K IS DIGITAL RESEARCH'S, PRIMOS IS PRIME'S, UNOS IS CHARLES RIVER DATA'S, IDRIS IS WHITESMITH'S LTD.

DOCUMENT PRODUCTION FEATURE

customized set of averages. Similarly, you can create personal lists of spelling words, sexist terms and awkward phrases for

diction to look for when processing your documents.

For information about the workbench itself, you can turn to


any of three programs. For a list of capabilities organized by command, you can use **wwbinfo**. To find commands organized by use, you can call on **help**. The **wwbaid** program, meanwhile, offers a range of lists and explanations.

The full Writer's Workbench set is barely more than a year old and AT&T Technologies has been marketing it for only a few months. As a result, the largest concentration of users today is still at Bell Laboratories, AT&T Technologies and a select number of software beta test sites.

Some testing of the results of using Writer's Workbench has been done at Bell Laboratories and several of the beta sites located at educational institutions. Although Bell presents these tests as evidence of the workbench's worth as a tool, a closer look at the testing shows that the results might at best be taken as anecdotal evidence. None of the tests done to date have used suitable control groups, and the evaluations have been generally performed by people with an interest in proving the efficacy of the programs. This, of course, is not to say the test results are false — only that the procedures, as described, were insufficient to be strongly indicative of anything.

Most commercial users I've talked to seem pleased with their results so far, but as of yet the majority are only familiar with the more prominent parts of the system. Surprisingly, the benefit most frequently cited by these users has been social. Many end users find it easier to accept criticism from the computer than from a co-worker or manager, while managers find it easier to suggest running a document through Writer's Workbench than to object personally to spelling and grammatical errors.

As for the package's individual functions, these users



NEW

ONE SIZE FITS ALL

Heurikon presents Minibox — a multiuser UNIX workstation based on its powerful HK68™ single board microcomputer and Uniplus+™ UNIX System III or System V operating system with Berkeley enhancements.

Designed with the OEM in mind, *one size fits all*. Both compact and flexible, the Minibox includes within its 10.5" w x 13.9" h x 20.5" l frame a 200 or 400 watt power supply, six slot Multibus™ card cage, (4-5 available for user use!), single double density floppy disk drive, streamer tape drive, and 31 or 65 Mbyte Winchester drive (expandable to 280 Mbytes). All this within the same cabinet! System status LEDs on the front panel inform the user of CPU and disk drive activity.

With Uniplus+™, Minibox becomes a flexible and affordable tool for program development, text preparation, and general office tasks. Included is a full "C" com-

piler, associated assembler and linker/loader. Optional languages are:

Macro assembler, ISO Pascal compiler, FORTRAN-77 compiler, RM-COBOL™, SVS BASIC (DEC BASIC compatible interpreter), SMC BASIC (Basic-Four BB3 compatible interpreter), and Ada™. Other utilities include UltraCalc™ multiuser spread sheet, Unify™ DBM, Ethernet™, and floating point processor. Alternate operating systems available are PolyForth™, Regulus™, CP/M 68K™, and others.

*UNIX is a trademark of Bell Laboratories. Unify is a trademark of Unify Corp. UltraCalc is a trademark of Olympus Software. Ethernet is a trademark of Xerox Corp. Uniplus+ is a trademark of UniSoft Corp. PolyForth is a trademark of Forth, Inc. Regulus is a trademark of Alcyon Corp. CP/M-68K is a trademark of Digital Research. Ada is a registered trademark of the U.S. government, Ada Joint Program Office. RM-COBOL is a trademark of Ryan-McFarland Corp. HK68 is a trademark of Heurikon Corp. Multibus is a trademark of Intel Corp.

MINI BOX

HEURIKON

© 1983

3001 Latham Drive
Madison, WI 53713
Telex 469532

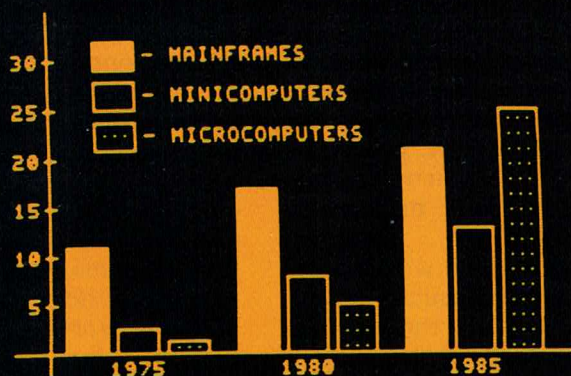
800/356-9602
In Wisconsin
608/271-8700

Circle No. 276 on Inquiry Card

Circle No. 82 on Inquiry Card ►

A Picture's Worth:

U.S. SHIPMENTS (\$ BILLIONS)



** GRAPHICS CAPABILITIES **

- ♦ HIGH PERFORMANCE GRAPHICS
- ♦ TEKTRONIX 4010 COMPATIBILITY
- ♦ AUTOMATIC SCALING (1023x1023) WITH 250x512 RESOLUTION
- ♦ ALPHA MODE (35 LINES x 73 COLS)
- ♦ OPTIONAL JOYSTICK
- ♦ CONNECTION TO LOW COST PRINTER FOR GRAPHICS HARDCOPY
- ♦ ASCII AND APL CHARACTER SETS
- ♦ BLOCK FILL, DOTTED/DASHED LINES

** DISPLAY TERMINAL CAPABILITIES **

- ♦ ANSI STANDARD CONFORMANCE
- ♦ DEC SOFTWARE COMPATIBILITY
- ♦ 80/132 COLUMNS, WINDOWING
- ♦ 4 PAGES OF MEMORY STANDARD (UP TO 8)
- ♦ 46 PROGRAMMABLE FUNCTIONS
- ♦ ASCII AND APL MODELS
- ♦ COMPACT ULTRA-THIN KEYBOARD
- ♦ HIGH RESOLUTION AMBER PHOSPHOR
- ♦ SETUP MODE



\$1895*

CONCEPT GVT+™ Graphics Display Terminal

*Quantity one, end user price; substantial discounts available.

human
designed
systems,
inc.

Whether used in video display mode or in its high-performance graphics mode, Human Designed Systems' GVT+™ Graphics Display Terminal offers more user friendliness, more design features, and more advanced functionality to optimize productivity — and encourage creativity — for the terminal operator, interactive user, and applications developer than any other terminal available today.



See us at
UNIX SYSTEMS EXPO/84
Booth 847

Atlanta — (404) 391-9763; Boston — (617) 449-6446; Chicago — (312) 825-2960; Dallas — (214) 437-1888; Delaware — Infocon: (302) 239-2942; Denver — (303) 469-1953; Hawaii — Gray Associates: (808) 261-3751; Houston — (713) 952-1403; Los Angeles — (213) 410-9454; Northern New Jersey — Infocon: (201) 624-1372; New York City Area — Infocon: (212) 689-8833; New York State — Naco Electronics: Rochester: (716) 223-4490; Syracuse: (315) 699-2651; San Francisco — (415) 692-4184; Washington, DC — International Systems Marketing: (301) 279-5775; Argentina — Iron SA: (01) 774-9369; Australia — Computer Clarity Pty. Ltd.: (02) 241 3385; Belgium — BELCOMP: 091-31.52.22; Canada — CAIL Systems: Toronto: (416) 362-1063; Denmark — ADCOM Data Aps: 1-19 44 66; Finland — Evumatic: 0 594 141; France — Walton: (1) 226.06.90; Japan — Ampere: 03 (365) 0825; Singapore — DTS Singapore: (65) 33-88-566; Switzerland — Mitek ag: 01/461 22 52; United Kingdom — Shandell Systems Ltd.: 02407-2027; Venezuela — H. Blohm SA: 2 541 21.22; West Germany — COMKO Computersysteme, mbH: 0221-48 30 51. INTERNATIONAL DISTRIBUTORSHIP INQUIRIES INVITED.

GVT+ is a trademark of
Human Designed Systems, Inc.

consistently mention **double**, **spellw** and **punct** as the most useful, while **diction** draws a mixed response. Bell Laboratories users place a strong emphasis on the readability (grade-level) scores, but other users seem to place more confidence in their own ability to set an appropriate pitch than in whatever advice Writer's Workbench might lend.

WHAT THE WORKBENCH DOES BEST

Most of the Writer's Workbench programs were originally developed for internal use at Bell Laboratories, so they make certain assumptions about their input text and environment. Almost all the programs that inspect text

require that the input be in a form ready for input to the **nroff** or **troff** text formatters. Furthermore, as delivered, most of the Writer's Workbench programs expect input documents to use either the **mm** or **ms** macro packages.

The programs work best, according to the manuals, on text closely resembling the original training materials and technical memoranda. In practice, this includes most expository writing – including academic papers, business reports and routine correspondence. The results offered by Writer's Workbench are not always useful with input documents not structured primarily as descriptive text paragraphs – for

example, procedural (step-by-step) instructions, lists or even poetry.

In their current state, the programs are not in any way represented to be artificial intelligence (AI) programs. Given the complexity of natural language and the basic simplicity of most of the programs, it would be unfair to expect perfect results. If you use Writer's Workbench, it can be helpful – but it definitely is no substitute for a careful reading of your text.

This is apparent in the manuals themselves, which presumably were run through the program during their development. Though the intent is always clear and the wording decipherable, you probably wouldn't want to

TRAINING

SYSTEM V

For 15 years, we've taught our own people to use the UNIX™ System. Now we can teach yours.

WHY AT&T FOR UNIX SYSTEM TRAINING?

AT&T offers the most current and comprehensive training on UNIX Systems.

AT&T provides the best learning environment; one terminal per student; evening access to facilities; and expert instructors.

AT&T has the breadth of courses your staff needs to unlock the full power of UNIX System V.

AT&T courses signal your commitment to improving productivity with high-quality training for your employees.

AT&T COURSES OFFER:

The same training and methods we use to

teach the UNIX System to our own people.

Rigorous classes designed to teach specific skills for job-specific applications.

Five areas of instruction ranging from introductory to advanced levels for Managers/Supervisors, Users, Systems Administrators, Applications Developers, and Systems Programmers.

Frequent class offerings so you won't have to wait for the courses you want.

Conveniently located training centers in Princeton, NJ; Columbus, OH; Lisle, IL; and Sunnyvale, CA. Or we'll bring our courses to your company and hold the training at your convenience.

For more information, a catalogue, or to register for classes, call 1-800-221-1647, Ext. 87.



LOGIC PROGRAMMING and ARTIFICIAL INTELLIGENCE

Silologic, Inc., a leading logic programming company developing artificial intelligence software, is seeking talented individuals to join its technical team. Specific areas of research and development include:

- *Unix*
- *Expert Systems*
- *Knowledge Management Systems*
- *Natural Language Processing*
- *User Environments*

We are seeking M.S. or Ph.D. level computer scientists, or software developers, with a knowledge of artificial intelligence research and experience with LISP or Prolog.

We offer exceptional compensation including profit sharing and stock options, and a highly technical development environment oriented towards individual achievement.

For further information, please contact:

SILLOGIC, INC.

6420 Wilshire Blvd., Suite 2000, Los Angeles, CA 90048
(213) 653-6470, Tlx TWX (910) 321-2963.

Circle No. 75 on Inquiry Card

An open-and-shut case for making MiniFrame™ your first choice in low-cost OEM systems.

ONE TO EIGHT USERS. MiniFrame can be configured for up to 50 Megabytes of integral mass storage: 5¼" Winchester fixed disks of 13, 26 or 50 Mb.

IMPRESSIVE CPU SPEED. Running the AIM™ Benchmark, MiniFrame is as fast as a VAX-11/750. The MiniFrame's MC68010 microprocessor operates at 10MHz, with no wait states.

VIRTUAL MEMORY MANAGEMENT. MiniFrame's custom MMU provides unique demand-paged implementation of UNIX™ System V... with 3.5 Mbytes of address space per process.

RUNS AS MANY AS 8 TERMINALS. Convergent PT and/or GT, or standard ASCII. Multidrop RS-422 communications line operates at 307 Kbit/sec.

FOR HIGH-SPEED PRINTERS, a Centronics-compatible parallel printer port is standard in the MiniFrame processor.

COMMUNICATIONS EXPANSION. LAN capability via optional high-speed Ethernet interface module, and/or 8 additional RS-232 ports.

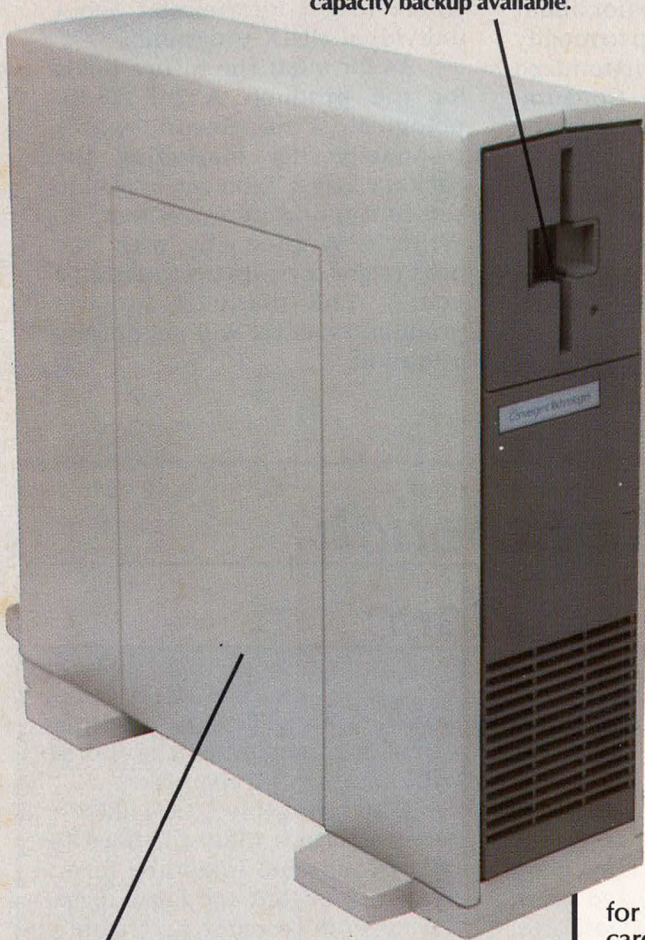
ECONOMICAL MEMORY EXPANSION. MiniFrame provides ½ Megabyte of RAM standard; up to three boards can be added for system expansion to 2 Mb.

MiniFrame: another first from Convergent Technologies. Super-minicomputer power at PC prices.

OEMs can now meet the needs of small to medium-sized organizations for low-cost, high-performance systems capable of handling large UNIX-based applications.

The MiniFrame system—which effectively opens a whole new market for you—is unique not just for its price/performance advantages. It also provides complete flexibility in meeting requirements ranging from single-user “personal UNIX systems” to eight-terminal data-processing installations.

BACKUP STORAGE with integral 5¼", 640 Kbyte floppy disk. Optional high-capacity backup available.



WE'VE CLOSED OUR CASE... now send for a complete information package. Write to Convergent Technologies, Data Systems Division, 3055 Patrick Henry Drive, Santa Clara, CA 95050. Phone: 408/980-0850. Telex: 176-825.

Convergent Technologies

Where great ideas come together

Convergent Technologies achieved this breakthrough by adapting the innovative hardware and software developed for its MegaFrame multiprocessor.

The result is a powerful, compact and expandable unit priced to give you the maximum opportunity of capitalizing on a rapidly growing UNIX market.

There are, of course, many other unique benefits of the MiniFrame system.

Convergent provides foundation software for office applications—including a powerful, Wang keystroke-compatible word processor, an advanced financial spreadsheet and complete electronic mail facility.

The Window Manager permits viewing and manipulating of up to four applications running simultaneously on Convergent's PT or GT terminal screens.

Major performance increases result from utilizing these terminals with the MiniFrame. At only slightly higher cost than standard “dumb tubes”, our PT or GT terminals offer high-speed communications plus built-in processor with enough memory to execute key portions of the system code.

Programming languages include industry-standard High Level COBOL and BASIC, full FORTRAN-77, Pascal and C.

OEM prices for the MiniFrame start at less than \$5,000; an eight-user MiniFrame can be configured for under \$10,000. Prices like these make it an open-and-shut case for choosing the MiniFrame.

Graceful upgrade path to the MegaFrame™

The MegaFrame, Convergent Technologies' revolutionary super-minicomputer system, utilizes multiple processors; has expansion potential to 128 users, 8 MIPS and gigabytes of disk storage. It enables OEMs to handle today's growing demand

for computing services without discarding a single piece of hardware... or being forced into expensive CPU upgrades.

Applications software can be transported—unchanged—from the MiniFrame to the MegaFrame whenever the workload requires it. The two systems are object-code compatible, allowing OEMs to provide a complete family of systems.

MegaFrame: proof that if anyone can build a UNIX system the way it should be built—it's Convergent Technologies.



MiniFrame and MegaFrame are trademarks of Convergent Technologies, Inc. UNIX is a trademark of Bell Telephone Laboratories, Inc., VAX is a trademark of Digital Equipment Corp., Ethernet is a trademark of Xerox Corp.

use the manuals as examples of good writing.

Similarly, using the programs regularly can be educational, but it's up to you to learn the lessons. For example, one sentence in a paper from a university professor at a beta test site contains nine different clauses, including five commas, three dashes and a colon.

Strangely enough, AT&T Information System's manual for its version of Writer's Workbench on the System 85 Applications Processor is considerably more graceful (this implementation of the workbench is also menu-driven and simpler to use).

Most of the Writer's Workbench programs were developed

at Bell Laboratories over the past decade. Pieces of it have been available for years. But AT&T Technologies is now marketing the complete package, supporting versions 2.0 and up running on UNIX System V.

Licensees who have purchased the source code (for about \$4000) may also sign up to modify the package and sell customized versions in binary form. Consumers may have to read some of the fine print on the new packages, though, because none of the licensees will be able to use the Writer's Workbench name. AT&T Information Systems is a licensee, as are several of the major software houses.

If you try to get information or

a copy of Writer's Workbench directly from AT&T, though, be prepared for some minor hassles. Many company people still seem confused about the division of responsibilities among the former Bell units in general, and they seem particularly foggy about where to get information about individual UNIX programs.

As for what the future holds for the product, AT&T Technologies' Wink Swain, who's responsible for marketing the package, says, "You can expect to see continuing improvements in Writer's Workbench, with the next major revision cycle within a year." The manuals for the product, he says, will get special attention. ■

Documenter's Workbench: Putting Text in its Place

Getting the right words down on paper is certainly important, but so is getting them down in the right places. AT&T Technologies has packaged a collection of the most-used UNIX formatting tools into a bundle it now sells as the Documenter's Workbench. In addition to the familiar printer and typesetter output formatters **nroff** and **troff**, the package includes two other formatters (**otroff** and **sroff**), three macro packages (**mm**, **man** and **mv**), several preprocessors (**pic**, **tbl**, **eqn** and **neqn**) and a few supporting utilities (**mmLint**, **checkmm** and **macref**).

In general, the formatters let you specify margins, headers,

footers, paragraph style, indentations and underlining. With the typesetting formatters (**troff** and **otroff**), you can choose font, type style, size and leading (interline spacing). With the right commands, you can add page numbering, tables, lists, footnotes, tables of contents and even indexes.

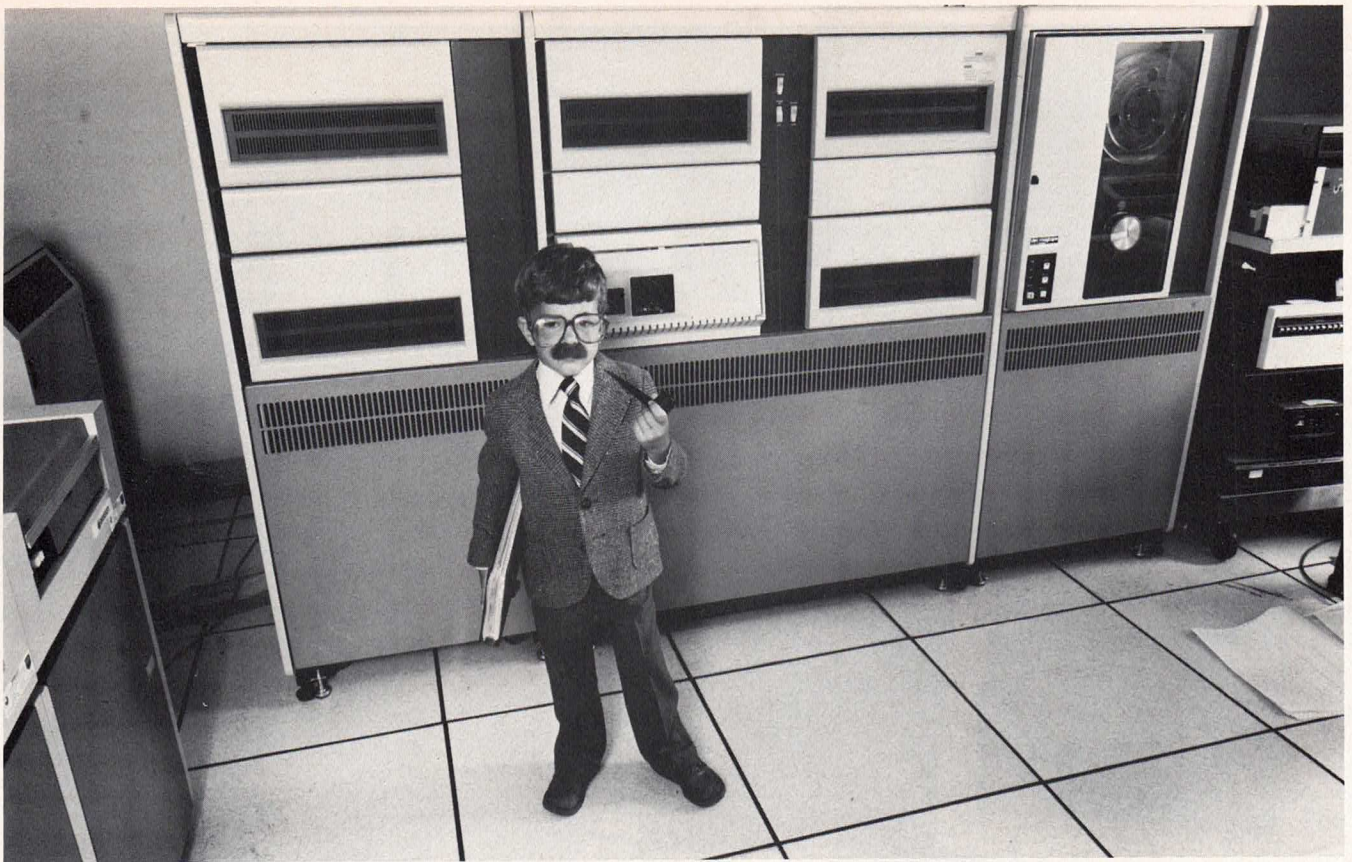
All the formatters and most of the macro packages take as input a textfile in which you've embedded lines containing formatting commands or a textfile that you submit with a command line containing the formatting options. Historically, UNIX editors have been strongly line-oriented, so it made sense to design the formatters to take input from

separate lines. That way, either text or commands could be altered without affecting the other.

Unfortunately, this interspersed format makes it hard to read documents in source form. This means that for most users the cycle is to enter text, run a formatting program, read the result and then go back to editing the source text. For people accustomed to dedicated word processing systems or even personal computer word processing programs, this can seem a long way around.

In return for editing and maintaining files in a form one or more steps removed from final form,

Continued to Page 90



WHAT YOU DON'T KNOW ABOUT UNIX* CAN STUNT YOUR GROWTH.

If your business depends upon computers and you *do not* know how UNIX will allow it to grow, your computer system could be threatened by obsolescence.

However, if you *do* know about the explosive emergence of UNIX as an industry standard for business, engineering, and scientific computers, in a multi-user and multi-tasking environment, you'll stand to reap extraordinary benefits.

Benefits that free you from the tyranny of dependence on a single hardware manufacturer. Protect your investments in software when you want to upgrade hardware. In short, the benefits of UNIX will grant you a freedom of choice that you never had before.

UNIX SYSTEMS EXPO/84 offers an unprecedented opportunity for business managers, information systems professionals, and all others with a need to better understand the full impact of UNIX. Over 150 of the industry's leading and most innovative companies will be exhibiting the latest hardware and software products. More than forty informative sessions will be offered and oriented to issues that are of prime concern to current and prospective users of UNIX-based systems.

The choice is yours. You can be content with your understanding of computer

technology as it is. Or you can learn everything you need to know about UNIX. And *grow*.

Act now for early registration discount. Send the coupon below prior to August 15th for a complete information package and pre-registration form.

Name	_____ 09
Title	_____
Company	_____
Address	_____
Telephone	_____
City	_____
State	_____ Zip _____

Complete and mail to: Computer Faire, Inc.
181 Wells Avenue, Newton, MA 02159
617/965-8350

UNIX* SYSTEMS EXPO/84

September 11-14, 1984 Los Angeles Convention Center

An exclusive production of Computer Faire, Inc.
A Prentice-Hall Company

*UNIX is a trademark of AT&T Bell Laboratories

Circle No. 77 on Inquiry Card

I N T E R V I E W



with **BILL JOY**

Bill Joy is one of those rare people who can carry on a rapid-fire technical conversation while coding at the keyboard. His seemingly inexhaustable energy has produced the C shell command interpreter, the vi screen editor and the Berkeley paging kernel, among other accomplishments. UNIX REVIEW sent Jim Joyce to Sun Microsystems, where Joy is Vice President in charge of Research and Development, to capture some of this energy.

REVIEW: *How did vi come about?*

JOY: It's an interesting story. I came to Berkeley in '75 as a theory student and got involved with Mike Harrison working on general context-free parsing algorithms, so I tried to write the thing in Pascal because Pascal had sets, which Ken Thompson had permitted to be of arbitrary length. The program worked, but it was 200 lines long - almost too big for the Pascal system. I talked to Thompson to figure out how I could make the Pascal system handle this program. Chuck Haley and I got involved in trying to

make the Pascal system handle it, but the thing was wrong because it was building the parse tree for the entire thing in core. So I got sucked in, got department help and built some hope of receiving enough support eventually to pay for this program to work under Pascal.

But while we were doing that, we were sort of hacking around on **ed** just to add things. Chuck came in late one night and put in open mode – where you can move the cursor on the bottom line of the CRT. Then George Coulouris from Queen Mary College came to Berkeley and brought along this thing called **em**, which stood for “editor for mortals.” It had two error messages instead of one. It had a prompt, and its own strange version of open modes done for ITT terminals, which really didn’t work very well on ADM 3As.

So Chuck and I looked at that and we hacked on **em** for awhile, and eventually we ripped the stuff out of **em** and put some of it into what was then called **en**, which was really **ed** with some **em** features. Chuck would come in at night – we really didn’t work exactly the same hours although we overlapped in the afternoon. I’d break the editor and he’d fix it and then he’d break it and I’d fix it. I got really big into writing manual pages, so I wrote manual pages for all the great features we were going to do but never implemented.

Eventually Chuck graduated with his Ph.D. for his part of the Pascal system. After he left, there was **ex** Version 0.1 at the Computer Center. That was a version of the editor from EP 016, which stood for September 1, ’76, the date that binary was created – after which we promptly lost the source because we were making so many changes and didn’t have **SCCS**.

Really, what started it all was that we got some ADM-3As to do screen editing. I remember right after Carter was elected, I was sitting in my apartment in Albany, CA, on a Saturday listening to people call Carter and ask stupid questions while I designed the

screen editor. That dates it: it was probably ’76. It was really a consequence of our initial frustration with Pascal. It went on from there. I stopped working on it whenever they made the reference cards – ’78 – ’79 – and I really haven’t worked on it for five years.

Mike Horton brought his editor along from Bell Labs called **hed**, for “Horton’s editor.” He was disappointed when **vi** won out over it. But **vi** had momentum with the local users – and Mark, somewhat out of frustration, went out and actually supported **vi**. That was nice, because I didn’t have the patience to do it anymore. Just putting the **termcap** entries in that people would mail me would take hours a week, and I was tired after three or four years.

REVIEW: *Didn’t Bruce Englar implement the count fields feature?*

JOY: Bruce suggested that. At one point there was an acknowledgment section in the documentation for the editor that mentioned all the people who had helped – I don’t know if it’s still in Volume 2.

A lot of the ideas for the screen editing mode were stolen from a Bravo manual I surreptitiously looked at and copied. Dot is really the double escape from Bravo, the **redo** command. Most of the stuff was stolen. There were some things stolen from **ed** – we got a manual page for the Toronto version of **ed**, which I think Rob Pike had something to do with. We took some of the regular expression extensions out of that.

REVIEW: *What would you do differently?*

JOY: I wish we hadn’t used all the keys on the keyboard. I think the interesting thing is that **vi** is really a mode-based editor. I think as mode-based editors go, it’s pretty good. One of the good things about EMACS, though, is its programmability and the modelessness. Those are two ideas which never occurred to me. I also wasn’t very good at organizing code when I wrote **vi**. I think the redisplay module of the editor is almost intractable. It does a really good

job for what it does, but when you’re writing programs as you’re learning.... That’s why I stopped working on it.

What actually happened was that I was in the process of adding multiwindows to **vi** when we installed our VAX, which would have been in December of ’78. We didn’t have any backups and the tape drive broke. I continued to work even without being able to do backups. And then the source code got scrunched and I didn’t

**I wrote manual
pages for all the
great features we
were going to do
but never
implemented.**

have a complete listing. I had almost rewritten all of the redisplay code for windows, and that was when I gave up. After that, I went back to the previous version and just documented the code, finished the manual and closed it off. If that scrunch had not happened, **vi** would have multiple windows, and I might have put in some programmability – but I don’t know.

The fundamental problem with **vi** is that it doesn’t have a mouse and therefore you’ve got all these commands. In some sense, it’s backwards from the kind of thing you’d get from a mouse-oriented thing. I think multiple levels of **undo** would be wonderful, too. But fundamentally, **vi** is still **ed** inside. You can’t really fool it.

It’s like one of those pinatas – things that have candy inside but has layer after layer of paper mache on top. It doesn’t really have a unified concept. I think if I were going to go back – I

wouldn't go back, but start over again.

I think the wonderful thing about **vi** is that it has such a good market share because we gave it away. Everybody has it now. So it actually had a chance to become part of what is perceived as basic UNIX. EMACS is a nice editor too, but because it costs hundreds of dollars, there will always be people who won't buy it.

REVIEW: How do you feel about **vi** being included in System V?

JOY: I was surprised that they didn't do it for so long. I think it killed the performance on a lot of the systems in the Labs for years because everyone had a copy of it, but it wasn't shared, and so they wasted huge amounts of memory back when memory was expensive. With 92 people in the Labs maintaining **vi** independently, I think they ultimately wasted incredible amounts of money. I was surprised about **vi** going in, though, I didn't know it was in System V. I learned about it being

command – little things like that. There were just dozens of people involved, but if you are in an environment where management says, "This person shall do an editor," it doesn't necessarily work. It's funny, the politics at Bell Labs.

REVIEW: You had said, when you were giving a demonstration earlier today, that when you are on foreign systems you use **ed**.

JOY: That's right. Absolutely.

REVIEW: You don't even try to use **vi**?

JOY: I'm used to having a 24-line terminal with no ability to scroll back. The reason I use **ed** is that I don't want to lose what's on the screen. I used to have a Concept terminal which had eight pages of memory, like a mini-version of a window system. I just don't like to lose what's in the window. I'm looking forward to the editor that is going to be embedded in the window system Warren Teitelman is working on. Having editing functionality everywhere

liked it. The problem was I spent all my time programming it because it was improving so fast that my programs kept breaking. I got tired of maintaining my macros so I guess I'm looking forward to an editor I can learn and then forget about.

I started to write a new editor not too long ago and had it about half done after two days. It was going to have almost no commands, but, instead use what's basically the smalltalk editing menu, a scroll bar and a thumb bar. Lines just went off to the right and if your window wasn't big enough – too bad – it just threw them away. There was going to be an edit file, and a store and read file. That was it.

It was called **be**. I'll let you guess what that stands for. It actually stands for about eight different things.

REVIEW: Bill's editor?

JOY: That's one of the eight. It's also the English verb "to be" because it is. There are six more.



The fundamental problem with **vi is that it doesn't have a mouse and therefore you've got all these commands.**

in System V quite a while after it had come out. They had this editor, **se**, but I guess it failed.

I think editors have to come out of a certain kind of community. You need a cultural context. As you mentioned, Bruce Englar thought of a number of things, Dick Fateman contributed work to the cursor position after the **join**

would be great in the same sense that it would be nice to have **history** everywhere.

REVIEW: So will there be a **history** mechanism in the new editor?

JOY: I would be surprised if there wasn't. Warren basically invented all those things. He's very keen on that. I tried to use EMACS and I

I got tired of people complaining that it was too hard to use UNIX because the editor was too complicated. Since I sort of invented the editor that was the most complicated, I thought I would compensate by also designing the editor that was the most simple. But I got distracted. If I had just spent another day on it.... I could

actually edit a file on it. I actually used it to edit itself and scrunched the source code – sort of old home day, because we used to do that all the time.

I had threatened to remove all the copies of **vi** on January 1 of this year and force people to use **be**. I don't think it would have worked, though, because I don't know any of the root passwords here anymore. These editors tend to last too long – almost a decade for **vi** now. Ideas aren't advancing very quickly, are they?

REVIEW: *So you use Interleaf now?*

JOY: I use Interleaf for all my documentation. When I'm writing programs, I can type them in half the time with **cat** because the programs are six lines – a **#include**, **main** and a **for** loop and one or two variables. I actually use **vi** for editing programs. James Gosling did a really nice editor as part of project at Carnegie Mellon University which is a AWYSIWYG: Almost What You See Is What You Get. It's also a program editor built into the window system he's working on. I think that will ultimately replace **vi**.

Interleaf is very nice. I expect there to be a lot of competition for programs like that. I don't expect that to be the only one. By the end of next year there will be half a dozen UNIX-based integrated office systems. Interleaf is based on the formatting process.

I think you'll see others focused on database, calendar management, mail, and spreadsheets – you need all these things to have a generic office automation application. I don't really know who is going to win. I know about a few that are unannounced, but it's not clear which is the most desirable. None of them are open, really. None are as programmable as UNIX. You really can't go in and add things that you need. That lack of programmability is probably what ultimately will doom **vi**. It can't extend its domain.

REVIEW: *Some would argue that **vi**'s domain is already far too extended.*

JOY: That's probably fair, too.

That's why it's so complicated, and has left and right parentheses commands. You start out with a clean concept and then sort of accrete guano. It lands on you and sticks and you can't do anything about it really.

REVIEW: *What is it that Interleaf offers you that EMACS doesn't?*

JOY: I can just look at my screen, and when I print it off, it's the same as it looks on the screen. It is formatted, and I'm tired of using **vi**. I get really bored. There have been many nights when I've fallen asleep at the keyboard trying to make a release. At least now I'll fall asleep with a mouse in my hand. I use the Xerox optica mouse instead of the other one because it is color coordinated with my office. Did you notice? I prefer the white mouse to the black mouse. You've got to have some fun, right?

This business of using the same editor for 10 years – it's like living in the same place for 10 years. None of us does it. Everyone moves once a year, right?

REVIEW: *What about Documenter's Workbench and Writer's Workbench?*

JOY: I used to use **diction**. I wrote some papers for some conferences and used **diction** on them. But with Interleaf I don't even have a **spell** program.

REVIEW: *Why?*

JOY: Don't need one. Well, I guess I do. I could use one. It just doesn't have **spell**.

REVIEW: *You don't use **spell**?*

JOY: I don't spell things wrong. Except t-h-i-e-r. But no, I don't generally have trouble with spelling mistakes. What **spell** did for me was catch errors introduced by the substitute commands. With sentences in **ed** or something, you see only one line, and substitutes can be done wrong. With **spell**, you can catch fuzballs that show up in your document. But **diction** is funny. I didn't like reading documents after **diction** got done with them.

REVIEW: *Did you use **suggest**?*

JOY: Yes, I've tried some of those things. I don't like reading things

that have been heavily dictionized because they don't flow. I would like to have an expert system that would help me but I don't think those (**style** and **diction**) are close enough. I don't need **double** or **spell**, either. I don't think any of those can help me write better or be better organized. I think an editor with a hierarchical sort of structure where I could look at the section outlines or make annotations in the margin would be helpful. Post-it notes are perhaps the greatest technological thing in the past 10 years. An electronic analog of the post-it would be wonderful so you could scribble on the document. I find much more of a need to just doodle on the screen than to run these programs. I think some of

**I think if I were
going to go back
I wouldn't go
back, but start
over again.**

these tools are overkill. Writer's Workbench is fine if you're stuck with **troff** and **nroff**.

I've never used **pic**. Some people have done some stuff with it, but it is too bad that instead of allowing you to think pictorially and draw pictorially, it forces you to translate images back into words and then compile back. That seems like the Linda Ronstadt song, "You're Taking the Long Way Around."

REVIEW: *You've already mentioned the mouse. What other hardware do you see for the documenter to make things better?*

JOY: I think the Macintosh proves that everyone can have a bitmapped display. The fundamental tension in UNIX that I think AT&T doesn't understand is that

everyone is going to have to have a bitmap. Bitmap display is media compatible with dot matrix or laser printers. With a mouse to point with, you've got sort of a baseline of facilities around which you can build a document environment. I think you also need a full-page display. I think we'll have to wait for Big Mac from Apple, maybe two years away, to get full-page display. I think a lot of the implications for developers is that this kind of development has to come from the low end, the Volkswagen of the document industry.

Document preparation systems will also require large screen displays. Something like the Sun is what I think you need – a 19-inch screen where you can see a full page and be able to put up screens and menus with something that's fast enough to allow you to scroll at a reasonable rate. We don't know how to do that without a mouse, really. All of the good research has been done using a pointing device.

Touch finger, joy stick, voice input are all either too late or too early.

REVIEW: *Voice is too early and touch is too late?*

JOY: I'm not sure voice yet works. I can't talk clearly enough. There was an editorial in *Datamation* about why the UNIX user interface is horrible. It was pretty poor, but the author does have some good things to say. I think he says something about people buying stoves. If you look at stoves and the way knobs are arranged. You'll see why it is that when you walk up to a random stove you can't tell which knob is going to turn on which burner. It's really stupid. There is no sensible way to put the knobs on the front to tell you. Some stoves have the knob in the center right next to the burners. That makes a lot more sense.

The point is that you want to have a system that is responsive. You don't want a car that talks to you. I'll never buy a car that says, "Good morning." The neat thing about UNIX is that it is very



**Ideas really aren't
advancing very
quickly, are they?**

responsive. You just say, "A pipe to B" – it doesn't blather at you that "execution begins," or "execution terminated, IEFBR14."

The trouble is that UNIX is not accessible, not transparent in a way that Interleaf is, where you sit down and start poking around in the menu and explore the whole system. Someone I know sat down with a Macintosh and a Lisa and was disappointed because, in a half hour, he explored the whole system and there wasn't as much there as he thought. That's true, but the point is in half an hour, almost without a manual you can know which button to push and you can find nearly everything. Things don't get lost. I think that's the key.

Systems are going to get a lot more sophisticated. Things will

tend to get lost unless the interfaces are done in the Macintosh style. People who use these machines may run applications but they won't necessarily be skilled at putting applications together. A lot of these people won't even have access to the underlying UNIX system.

The fundamental tension in System V is that it is oriented toward a character-mapped environment. The software you have to build is completely different. You don't assume a mouse and you don't assume a reasonable-sized display. You just forget it. Those are two different problems.

It's ultimately the media and the set of peripherals that you have on your machine that affects what the user sees. I don't think the Macintosh software is of any value. I'm not even sure it can be taken to a larger machine. You can spend your time making software small, or you can spend your time making it functional and sensible. You can't do both. I think there is an axe that is going to chop the two apart.

You'll see WordStar, the database sort of word processing environment that doesn't have bitmaps, and you'll see the ones that do, and the difference between the two will be like night and day... The Macintosh's days are numbered. Non-bitmap machines have no future. Personal laser printers will see to that. The days of non-raster stuff are numbered, though sheer momentum will carry it to the end of the decade. These things come and go.

We went from printing terminals to dumb CRTs to smart CRTs, with tangents off toward storage tube displays and black and white bitmaps. I think the days of even black and white bitmap are very numbered. Color will take care of that. And then, with the demise of the last vacuum tube, which is the CRT, and with the advent of thin film transistors, which will be flat displays, it will all be color.

Black and white bitmaps supplanting CRTs make for a small



EMERALD ONE™

SOFTWARE WITH COURAGE, BRAINS AND HEART

WHAT IS EMERALD ONE?

The most complete integrated office system available today, EMERALD ONE combines the most essential office tasks through six fully compatible and seamless sets of tools. EMERALD ONE runs on a broad range of mainframe, mini, super-micro and personal computers which use the UNIX™ operating system—the emerging standard for the office.

THE TOOLS OF EMERALD ONE

EMERALD ONE integrates your office tasks through:

1. *COMMUNICATIONS*, including Telephone Messaging and Electronic Mail systems,
2. *INFORMATION HANDLING* with EMERALD ONE's powerful Relational Database system,
3. *DECISION SUPPORT* features such as Business Graphics and the Electronic Spreadsheet,
4. *DOCUMENT PREPARATION* with Word Processing and a Cabinet, Document and File Folder system,
5. *TIME MANAGEMENT* tools such as the Personal Diary system and Meeting Scheduler and
6. *SYSTEM ADMINISTRATION* functions that allow a non-technical user to customize EMERALD ONE for the individual, work group and organization with ease.

SOFTWARE FOR THE WORK GROUP

EMERALD ONE goes far beyond stand-alone personal computer software by linking individuals and their work groups. With EMERALD ONE, users work as a communicating group, not as isolated individuals. Whether it be a document, spreadsheet or personal diary entry, everything created with EMERALD ONE can be exchanged easily between individuals, work groups and beyond.

EMERALD CITY, THE PEOPLE BEHIND EMERALD ONE

EMERALD ONE is the result of an intensive, multi-year research commitment by Emerald City and its sister company Trigon Systems Group, one of the most respected consulting companies in office integration. Attractively priced for distribution by hardware manufacturers, system integrators and OEM's, EMERALD ONE is fully supported by an extensive marketing program designed to assist distributors in penetrating the integrated office market. Emerald City offers the reality of a complete business solution, not just technology.

Emerald City, the company with courage, brains and heart.

EMERALD
C · I · T · Y

Emerald City Inc.
20 Richmond Street East, Suite 700
Toronto, Canada M5C 2R9
(416) 863-9923



UNIX is a trademark of Bell Labs
EMERALD ONE is a trademark of
Emerald City

The days of non-raster stuff are numbered, though sheer momentum will carry it to the end of the decade.



wavelet, but if you don't see that little wavelet, you're really going to get hit by the tsunami that is to come.

I've wiped **troff** off my machine, and I'd rather live with the bitmap world than in the **spell/diction** world. I want to get mud in my face and arrows in my back with the bitmap.

REVIEW: *The basic UNIX tools that can be used for documentation are plentiful but misunderstood. For example, the use of **make** to do documentation control. What are your views on that?*

JOY: I think **make** is the program that causes people to write the things down that formerly were scribbled on the wall. It's sort of the graffiti recorder. That's the wonderful thing about it. People don't use **SCCS** and **make** enough. The people here doing documentation now use **SCCS**, mostly because I put all documentation under **SCCS** and sort of twisted people's arms into using it.

REVIEW: *Real programmers use **cat** as their editor.*

JOY: That's right! There you go! It is too much trouble to say **ed**,

because **cat**'s smaller and only needs two pages of memory – plus you're not likely to get swapped out. That's why **ed** didn't prompt, you know. The performance of the system then was just horrible. It would swap things out randomly and do all sorts of things. In **ed** you might type "a", but have no idea how far behind the system was. And it didn't matter, as long as it didn't get more than a few hundred characters behind and start throwing lines away without telling you.

Typically it wasn't that bad. If it had been prompting, you would hit carriage return and wait for the prompt, and it would have taken three seconds to comment. That's something we noticed when we put **em** up. We put in the prompt and suddenly realized it had to go through the operating system.

I think UNIX has lived with grace for years. We've had the grace of people not being able to tell when the system was doing a bad job of scheduling the CPU. Now we can't hide behind time-sharing.

I think **SCCS** is misunderstood. I think **make** has never had

a good document. Henry McGilton just finished rewriting a **troff** manual for the first time. Since **troff** has never really had a manual, he had to sit down and figure what some of these things meant – backslash, right adjusting tab stops. No one ever really wrote a good manual for it – partially because Joe Osanna, who wrote **troff**, died in 1976. The program was written in assembly language, then translated line for line in C and it's all done with global variables – it's an ancient program. It's basically an accretion of all this completely unrelated stuff on top of a very, very small base. It's not surprising people don't understand it.

When you look at the manual, **tbl** looks really good but you sort of get it right by iterative approximation; it's very difficult to get a good looking table. I think the thing that's really missing is that none of these things help you with graphics or graphic design very much. I want a program that helps me learn how to draw and learn how to paint. Some attempt is made at that by **pic** but it's solving it in the wrong domain. I

don't want to type "arc from A to B." I wouldn't mind saying that, though. Maybe that's the answer: talk to the program.

I think the hard thing about all these tools is that it takes a fair amount of effort to become proficient. Most people are lazy. I'm lazy. I'm enjoying using other people's software now. At Berkeley for so long, all the software we were using was stuff I had something to do with and that wasn't any fun. I have fun with Interleaf because if it crashes, I don't feel responsible. I've even divested myself of responsibility for the operating system. I don't have to worry about that crashing. Editing without guilt.

REVIEW: *All the directions you were talking about really assume a lot more compute power than we have at present.*

JOY: I think that to make that assumption is bad. All projections that I see have memory going at \$300 a megabyte by 1989. Soon the processor will be \$50, and you'll be able to use it to refresh video. There are too many good things you'll be able to do with this stuff for it not to be available cheap. The real cost is very low. One has to wonder what software

the phone off, thank you. I think that's going to require very different technology.

REVIEW: *You mention everything but disks.*

JOY: You might want to page over satellite telephone... Page fault, and the computer makes a phone call. Direct broadcast satellite or audio disk – that's the technology to do that. It's half a gigabyte – you get 100 kilobyte data rate or a megabyte or something, I don't remember. You can then carry around with you all of the software you need. You can get random data through some communications link. It is very like Dick Tracy. Have you seen these digital pagers? You can really communicate digital information on a portable.

I don't think you need to have a disk with you. There are so many people who believe that you need to have a disk that you'll be able to have one because they'll make it cheap. That's the way things work. It's not what's possible but what people believe is possible. That's what makes imagination so wonderful, right? Silicon is such an incredible amplifier. If you can figure out what should be and you get people to believe you enough that they will give you money, you can almost make it come true. That's why bubble memory has never made it. People didn't believe it was the right thing to do. But there's nothing wrong with bubble memory.

There's an incredible amount of momentum in the technology. Look at the momentum in vacuum tubes. It's all an economy of scale. There's an incredible momentum in UNIX. It really doesn't matter what UNIX is anymore. It ceased to matter when the vendors started adopting it. People used to call me up saying, "I don't know what UNIX is but I've got to have it! How do I get it?" It's at that point now.

REVIEW: *Like designer jeans?*

JOY: I don't buy designer jeans – well, what I'm wearing aren't bad jeans. They're my burlap sacks. Wrinkles are in, you know. ■

**If you don't see
that little wavelet,
you're really going
to get hit by the
tsunami that is
to come.**

is going to be worth, too. It's going to be produced in such enormous volume.

When can that stuff go portable? You don't really want to have a telephone in the office or be tied to an office. You'd like to have the office with you and the phone with you. I want to be able to turn

Take the **RELIABLE**
Approach!

**MCBA™ Software Now Available
on Microcomputers**

- NCR Tower 1632
- Codata 3300
- Plexus P/(all)
- General Ledger
- Payroll
- Receivables
- Payables
- Order Processing
- Purchase Orders
- Inventory
- Bill of Materials

For Business Software, call us

RELIABLE
DATA SYSTEMS

**Full Function Full Feature
Fully Integrated**

- RM Cobol
- Source Code Available to Dealers
- Multi-User and Shared Files
- Multi-Company Data Bases
- Many run standalone
- Add others as needed
- Field Proven
- Full Documentation

**900 San Antonio Road
Los Altos, CA 94022 (415) 949-3600
Dealer Inquiries Welcome**

MCBA is a registered trademark of
Mini Computer Business Applications, Inc.

Circle No. 297 on Inquiry Card

UNIX* **JOBS** **REGISTRY**

National registry of candidates and jobs in the Unix field. Please give us a call; send a resume; or request a free Resume Workbook & Career Planner. We are a professional employment firm managed by graduate engineers.

800-231-5920

P. O. Box 19949, Dept. UR
Houston, TX 77224
713-496-6100



Scientific Placement, Inc.

*Unix is a trademark of Bell Labs

Circle No. 225 on Inquiry Card

INDUSTRY INSIDER

Of mice and windows

by Mark G. Sobell

Imagine a work station with a large bitmapped screen, a mouse and flexible windowing capabilities. You log onto your local system, open a window on the screen, and pass through your system to log onto another remote system. Your connection to the remote system is made via a combination of electrical wires and fiber-optic conduit. You read network mail on the remote system and locate a paragraph you'd like to quote in a document you are preparing on a third system.

You open a second window on the screen, log onto that third system and start editing your document.

Using only the mouse, you move the cursor back to the first window and select the paragraph you want. Then you return the cursor to the second window and deposit the paragraph where you want it in your document.

Sound fantastic?

What about a 10-page report, due in an hour, that you just learned needs to be typeset in a larger font, formatted to fit on a smaller page than you initially prepared for and changed to feature pie charts in place of the horizontal bar charts you used?

Sound like a problem?

OPTIONS AVAILABLE TODAY

Using software available now, you could move the paragraph and have your report today.

I had the pleasure of visiting Sun Microsystems recently and meeting with John Gage, director



of the Sun science office. John showed me some of the ways Sun is preparing for the future with Workstation Publishing Software (tm) written by Interleaf, a Massachusetts-based software company. He played out just the scenario I outlined above, transferring text from one remote system to another. All the systems were running UNIX (4.2 BSD) and were linked by Ethernet.

Even the "little" workstation system Gage had under his desk had 2 MB of memory and an Ethernet and graphics controller. But it didn't have a disk or tape. Any time the system got a command, it asked the host machine to download the appropriate program — and it did so quickly.

If you've never seen a bar chart turned into a series of pie charts in the twinkling of an eye, let me assure you — it's quite a sight. The charts take form as if

they were painted by a high-speed second hand. For six charts, it takes about 10 seconds for the complete conversion!

Interleaf touts what it calls a "structured document editor" that operates in real time. Using this editor, you can change the characteristics of a single component (paragraph, heading or title, for instance) or change the characteristics of an entire group of components at once. Modifiable characteristics include margins, indentations, font styles, type sizes, margin justifications and tab settings. Any change you make is reflected on the screen almost instantaneously. In the demonstration, the type size of a 35-page document was changed, with all attendant repagination, in the course of about 15 seconds.

You can perform all aspects of document control with a mouse, which should be quite easy to use with a little practice. When you point to a component and push a button on the mouse, the software displays a "property sheet" that lists the characteristics of the component. You can then use the mouse to make changes in the property sheet.

When you point to a chart using the mouse, the software displays the parameters of the chart. You can change the form of the chart, revamp its scaling, add new shading or change the data upon which it is based. You can also look at a "wishbook" that allows you to see the different kinds of charts you can use (for example: bar, pie or line).

THE CODATA DIFFERENCE

A Major Improvement in OEM Microcomputers.

With so many microcomputers on the market, and with each manufacturer claiming to have the solutions to everyone's problems, why are more and more OEM users choosing CODATA?

Very simply because the CODATA 3300 works!

That's an important part of what we call THE CODATA DIFFERENCE—a refreshing, new concept in OEM marketing.

More performance... Lower price... Extended warranty.

We've increased the performance of the CODATA 3300 and, at the same time, we've lowered the price. That means our CODATA multi-user, multi-tasking 16-bit computer systems have price/performance ratios nobody else can match. We're so confident in our system's integrity that we've extended the warranty on every CODATA 3300 to 180 days. That's the longest war-

ranty in the business and, let's face it, you've got to be pretty sure of your product to guarantee it for six full months. We're sure.

These are more elements of THE CODATA DIFFERENCE.

An uncomplicated pricing policy.

We have only two price sheets: one for suggested retail, the other for OEM prices. Simple, straightforward, easy-to-understand. Beyond that, there are no quantity commitments, no bill-backs, no hassles.

Full support and technical assistance if needed.

Our customers select the CODATA 3300 because it fulfills their needs, because it does what we say it will do, and because we have taken great pains to assure that it is functionally simple and capable of performing its

assigned tasks with no problems. But, if technical support is required it's only as far away as your phone and available at a moments notice via our TOLL FREE number during normal business hours in every U.S. time zone.

A multi-user, 33-megabyte system for only \$9,600.

The CODATA 3300 is a powerful 16-bit, 68000-based, MULTIBUS system that can effectively accommodate up to ten users. It's a complete UNIX system that runs full ANSI standard FORTRAN-77, RM/COBOL, BASIC+, SMC BASIC, APL, and PASCAL. The 3300 provides up to 33 megabytes of unformatted on-line storage via an integrated, high speed Winchester drive, and a removable, quad-density 5 1/4" floppy disc system. The 3300 features 320 Kbytes of parity protected RAM.

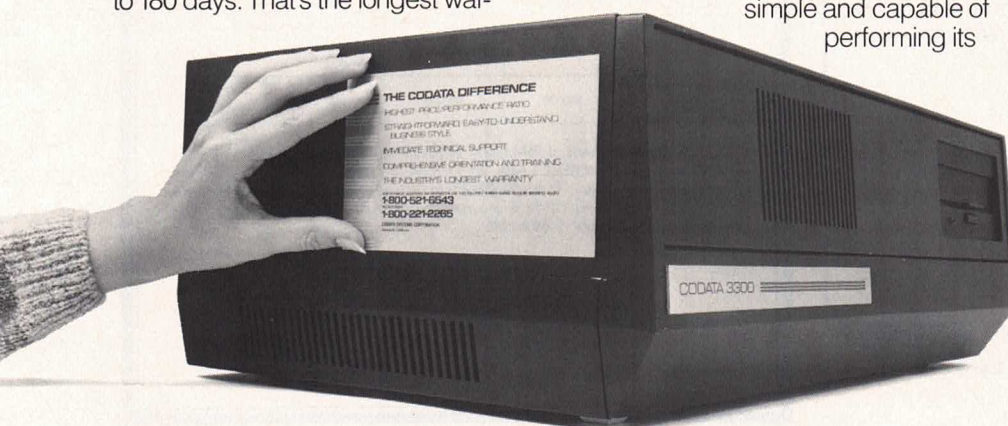
An expanded CODATA 3300, with 84 megabytes of disc storage, is priced at \$13,500. The 3300 is also available as a 12-megabyte system for as low as \$7,800.

LET US SHOW YOU THE CODATA DIFFERENCE.

Write or call us for more information.

CODATA SYSTEMS

285 North Wolfe Road
Sunnyvale, CA 94086
1-800-521-6543
In California: 1-800-221-2265



When you point to a paragraph, the property sheet displays information about the format of the paragraph (type size, type style, indentations, tabs, etc.).

The Workstation Publishing Software also includes a graphical objects editor that allows you to design your own line art. Using this editor, you can create, resize and duplicate a line drawing. You can also fill the drawing with various textures and position labels and titles as needed.

Excellent communications between remote machines and high-powered word processing capabilities are becoming more and more common, and will ultimately be required of products serving offices of the future.

THE FUTURE

UNIX is a standard array of tools that can be easily put together to build the system of

programs you need. The office of the future will present larger, more powerful and more complex tools. But the tools will be easier to use than they currently are because much time and money has gone into creating user interfaces designed for people.

No longer will you need to pipe output from **nroff** through **pr** to **lpr** to obtain dot matrix output. Rather you will be able to use windows and mice to review material by pictorially moving pages of a document from one machine to another. The machines will process the pages with programs that check the text (spelling, punctuation and sentence structure), integrate graphics and numerical data, display the output on a high resolution screen and produce laser-generated or photo-typeset hard copy when needed.

As an example, Interleaf says that its software will accept input from Multiplan, Visicalc and Lotus

1-2-3. Thus, an accountant can put together a financial projection using a spreadsheet package and integrate the projection (in the form of a chart) with text to produce a typeset report.

This sort of integration is the basis of UNIX and is essential in order to realize the potential of the powerful hardware and software available today.

Because the purpose of computers is to organize and provide timely information, and because it is unlikely that any one program will be able to provide everything every user needs, the designers of the new UNIX tools must maintain the tradition of tools that can work together.

With this communication between programs, system users will be able to take fullest advantage of each vendor's expertise while minimizing time-consuming and potentially error-producing re-entry of data. ■

MIPS SOFTWARE PROVIDES

THE APL - UNIX® SOLUTION:

DYALOG APL

UNIX® based - fully functional commercial APL including nested arrays, upper and lower case data support for the UNIX® environment, dynamic workspace size, external functions (callable subroutines written in other languages) full screen editor, error trapping, commercial formatter and a host of other desirable features.

DYALOG APL is available for a variety of UNIX® computing environments including VAX®, PE, Gould, 3 B Series!, NCR Tower, Zilog, Fortune, Perq, Ridge!, Pyramid!, and Sun!. For further information about DYALOG APL on your 68000, 16032 or 8086 base system, call or write today.

MIPS SOFTWARE DEVELOPMENT, INC.
31555 West 14 Mile Road
Suite 104
Farmington Hills, MI 48018
313-855-3552

UNIX® is a trademark of Bell Laboratories.
VAX® is a trademark of Digital Equipment Corporation.
1. Call for Availability

α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω

Circle No. 283 on Inquiry Card

C COMPILER

- FULL C
- UNIX® Ver. 7 COMPATABILITY
- NO ROYALTIES ON GENERATED CODE
- GENERATED CODE IS REENTRANT
- C AND ASSEMBLY SOURCE MAY BE INTERMIXED
- UPGRADES & SUPPORT FOR 1 YEAR

C SOURCE AVAILABLE FOR \$2500⁰⁰

HOST	6809 TARGET	PDP 11* LSI 11* TARGET	8080 (Z80) TARGET	8088 8086 TARGET
FLEX* UNIFLEX* OS-9*	\$200.00 \$350.00 <small>(11/11/83) (11/11/83) (11/11/83)</small>	500.00	500.00	500.00
RT 11* RSX 11* PDP 11*	500.00	200.00 350.00 <small>(11/11/83) (11/11/83) (11/11/83)</small>	500.00	500.00
CP M* 8080 (Z80)	500.00	500.00	200.00 350.00 <small>(11/11/83) (11/11/83) (11/11/83)</small>	500.00
PCDOS* CP M86* 8088 8086	500.00	500.00	500.00	200.00 350.00 <small>(11/11/83) (11/11/83) (11/11/83)</small>

*PCDOS is a trademark of IBM Corp. MSDOS is a trademark of MICROSOFT. UNIX is a trademark of BELL LABS. RT 11/RSX 11/PDP 11 is a trademark of Digital Equipment Corporation. FLEX/UNIFLEX is a trademark of Technical Systems consultants. CP/M and CP/M86 are trademarks of Digital Research. OS-9 is a trademark of Microware & Motorola.

408-275-1659

TELECON SYSTEMS

1155 Meridian Avenue, Suite 218
San Jose, California 95125

Circle No. 23 on Inquiry Card

C ADVISOR

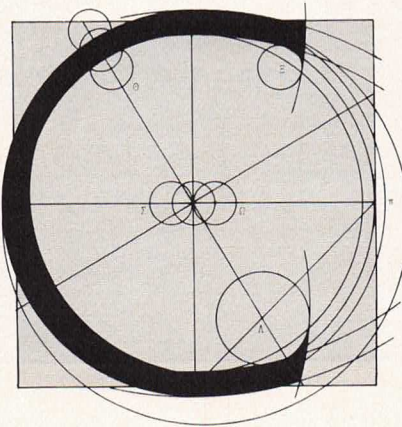
Screen handling from C programs

by Bill Tuthill

Programmers who come to UNIX from CP/M or MS-DOS are often surprised to find that UNIX provides no easy way to write full screen software. For example, when writing help menus for self-documenting programs, you might want to paint the screen with options that users could choose among by entering numbers or moving the cursor to certain locations on the screen. Each help menu would then form a separate panel. But UNIX is normally a scrolling, rather than panel-oriented, system.

Scrolling systems move the display up one line each time there is a new line of input or output at the bottom of the screen. Panel-oriented systems, on the other hand, treat the screen as a fixed sheet of paper. Generally, output replaces data already on the screen, and input is made by entering one character chosen from a menu. Lotus 1-2-3 is an example of a panel-oriented system. Under UNIX, when you want to read a single character rather than an entire line, you have to change the terminal to an alternate mode! This article explores one method of screen handling that suffices for writing help panels.

There are two ways to write full screen UNIX software in a terminal independent way. The first is with **termlib(3)**, a small library of low-level routines that accomplish cursor motion and screen output. The second is with



curses(3), a package of medium-level routines for cursor movement and screen updating. For simple applications, **termlib** is preferable because of its stability and efficiency. A future column will discuss the **curses** package in detail.

With 4.1 BSD, the name **termlib** was changed to **termcap**. This is confusing because the terminal capability file, **/etc/termcap**, is altogether different, even though it shares the same name. The **termcap** file is a readable ASCII database, while **termlib** is a library of C routines in **/usr/lib/libterm.a**. On System V, **termlib** is the only name given to the C library package discussed here. Therefore, for the sake of clarity and portability, the name **termlib** will be used.

The **termlib** library package is composed of six routines. The most important one, **tgetent()**, reads the appropriate terminal

description from the **/etc/termcap** database into an array declared in your program. The address of this array is known to the other routines because it is stored as an external variable, declared inside the library. Hence, your array must be retained through subsequent calls to the remaining five library routines.

Terminal capabilities as defined in the **termcap** database can be divided into three groups: boolean, numeric and string functions. Boolean capabilities are either true or false; for example, **:am:** states that a terminal has automargins. Numeric parameters give a value to some terminal characteristic; for instance, **:co-80:** means that a terminal has 80 columns. String functions contain a terminal control sequence to perform some operation on the screen; for example, **:cl=^Z:** indicates that CTRL-Z will clear the display. These three groups of capabilities are extracted with the **tgetflag()**, **tgetnum()** and **tgetstr()** routines, respectively. A capability ID must be passed as a parameter to these routines. One such capability ID is "cl" for the clear screen string, used in the program accompanying this article. Finally, the **tgoto()** routine generates cursor motion sequences and the **tputs** routine actually sends output to the screen. Here is the manual page description of the six routines:

```

tgetent(entry, term)      /* get termcap entry from /etc/termcap */
char *entry, *term;

tgetflag(id)              /* get boolean capability from entry */
char *id;

tgetnum(id)               /* get numeric capability from entry */
char *id;

char *
tgetstr(id, area)         /* get string capability from entry */
char *id, **area;

char *
tgoto(cm, col, line)      /* generate cursor motion sequence */
char *cm;
int col, line;

tputs(cp, nlines, outc)   /* send output to the screen */
register char *cp;
int nlines;
int (*outc) ();

```

The six **term**lib library routines.

Note that all routines return an integer, except **tgetstr()** and **tgoto()**, which both return character pointers. Notice, too, that the **tgetstr()** routine must be handed a pointer to a character pointer. This variable, **area**, should begin by pointing at a character array. Each time **tgetstr** is called, it will move the pointer along, filling in the character array with appropriate terminal control sequences as it proceeds. It will terminate each string by adding a **NULL**. When you ask for "cl" on a Zenith z29 terminal, for instance **tgetstr** will fill in the array with: **ESC E NULL**. Each time, **tgetstr()** will return a pointer to the beginning of the string it has just filled in. This arrangement provides an efficient and compact method for storing string capabilities.

Next on the manual page listing above, the **tgoto()** routine must be passed the cursor motion string, derived by **tgetstr()**, along

with the destination line and column. It then produces a cursor addressing string, which can be sent straight to the terminal. When you ask for "cm" on an Ann Arbor Ambassador, for example, **tgetstr()** will yield the string **E%i%d;%dH** for cursor motion. This string by itself will not have the desired effect of moving you to a particular location on the screen. If you want to move to column 20, line 5, you need to call **tgoto(CM,19,4)** – because the numbering of columns and lines starts with zero. On an Ann Arbor Ambassador, this would yield the string **ESC 20 ; 5 H** (no spaces, of course), which would move the cursor to the place you want it. The actual output of cursor addressing sequences must be done with **tputs()**, because the package needs to supply padding information if necessary. The routine **tputs()** needs to know how many lines are affected so it knows how much padding is necessary. You

also have to supply a character output routine. On the manual page listing above, this routine is called **outc()**, and in the sample program included later in this article, it is called **putchr()**. That's the theory, in any case. Now for a practical example.

A SAMPLE PROGRAM

The program on page 72 calculates monthly mortgage payments. This can be done with just a few lines of code but this particular program was designed to demonstrate the use of **term**lib. The source code must be compiled with two libraries – one for handling the terminal and one for handling the math library, since we will need to do exponentiation:

```
% cc mortgage.c -o mortgage -lterm -lm
```

The **putchr()** routine is included because we have to supply **tputs()** with a real output function. Since **putchar()** is a preprocessor macro, defined in

THE WORLD'S MOST POPULAR

UNIX SYSTEM

FOR THE

IBM PC

AND

LISA 2

Now, SCO brings the world's most popular version of AT&T's UNIX™ Operating System — and a comprehensive line of applications software — to the IBM® Personal Computer™ and the Apple® Lisa 2™. XENIX™ from SCO is simply the best microprocessor-based UNIX you can buy. **No one else in the UNIX business provides you with better support, and no other version of UNIX gives you all these features:**

VISUAL SHELL:

The XENIX visual shell provides you with a screen-oriented, menu-driven interface—with selectable color for the IBM PC.

MULTIUSER:

With XENIX, 3 or more users may efficiently use a single IBM PC or Lisa 2 simultaneously.

MOST APPLICATIONS:

With nearly 80% of all micro-based UNIX systems running XENIX, more applications are available for XENIX than for any other version of UNIX.

MULTISCREEN™:

SCO's Multiscreen™ allows you to run multiple programs simultaneously and view each program's screen with a touch of a key.

MOST COMPLETE:

XENIX is the most complete version of the UNIX System available for micros—including Microsoft's commercial enhancements and key extras from Berkeley 4.2 UNIX.

PC-DOS:

XENIX lets you read and write PC-DOS files on the IBM PC—and a cross-development environment will be available soon.

DOCUMENTATION:

SCO's XENIX documentation is written to meet the needs of the small systems marketplace — well-organized, clear, concise.

The powerful family of XENIX systems software and applications is available today from SCO — your UNIX Systems Software Company since 1978 — to set a new standard for personal computer productivity.

UNIX is a trademark of Bell Laboratories.
IBM is a registered trademark and IBM PC, IBM Personal Computer are trademarks of International Business Machines Corp.
Apple is a registered trademark and Lisa 2 is a trademark of Apple Computer, Inc.
XENIX is a trademark of Microsoft Corp.
Multiscreen is a trademark of The Santa Cruz Operation, Inc.
© 1984 The Santa Cruz Operation, Inc.

SCO
THE SANTA CRUZ OPERATION

YOUR UNIX SYSTEMS SOFTWARE COMPANY

500 CHESTNUT STREET
P.O. BOX 1900
SANTA CRUZ, CALIFORNIA 95061

408/425-7222

Let's go to work.

```

/*
 * compile with: cc mortgage.c -o mortgage -ltermcap -lm
 */
#include <stdio.h>
#include <math.h>

putchr(c)                /* tputs can't handle the putchar macro */
register char c;
{
    putchar(c);
}

main()                    /* mortgage - payment calculation program */
{
    char    *term, entry[1024], buf[512], *area;
    char    *getenv(), *tgetstr(), *tgoto();
    char    *CL, *CM, *SO, *SE;
    double  principal, interest, payment;
    int     years;

    term = getenv("TERM");                /* get tty type from env */
    if (tgetent(entry, term) != 1)        /* termcap entry for tty */
        perror("bad termcap"), exit(1);
    if (tgetnum("li") < 10)                /* check number of lines */
        perror("too few lines"), exit(1);
    if (tgetnum("co") < 40)                /* check number of cols */
        perror("too few cols"), exit(1);
    area = buf;                           /* ptr to strings in buf */
    CL = tgetstr("cl", &area);             /* clear screen string */
    CM = tgetstr("cm", &area);             /* cursor motion string */
    SO = tgetstr("so", &area);             /* begin reverse video */
    SE = tgetstr("se", &area);             /* end reverse video */
    tputs(CL, 1, putchr);                  /* now clear screen */
    /*
     * now ask for the 3 things we need: principal, interest, term
     */
    tputs(tgoto(CM, 21, 4), 1, putchr);    /* line 4 column 21 */
    printf("%sprincipal%s: $", SO, SE);    /* ask for loan amount */
    scanf("%lf", &principal);
    tputs(tgoto(CM, 22, 5), 1, putchr);    /* line 5 column 22 */
    printf("%sinterest%s: %%", SO, SE);    /* ask for interest rate */
    scanf("%lf", &interest);
    tputs(tgoto(CM, 18, 6), 1, putchr);    /* line 6 column 18 */
    printf("%sno. of years%s: ", SO, SE); /* ask for loan period */
    scanf("%d", &years);
    /*
     * now calculate monthly payments using pow(3), print answer
     */
    interest /= 100;                       /* in percent */
    payment = principal * ((interest / 12) /
        (1. - pow(1. + (interest / 12), -((double)(years * 12)))));
    tputs(tgoto(CM, 15, 8), 1, putchr);
    printf("%smonthly payment%s: $%.2f", SO, SE, payment);
    tputs(tgoto(CM, 0, 10), 1, putchr);
    exit(0);
}

```

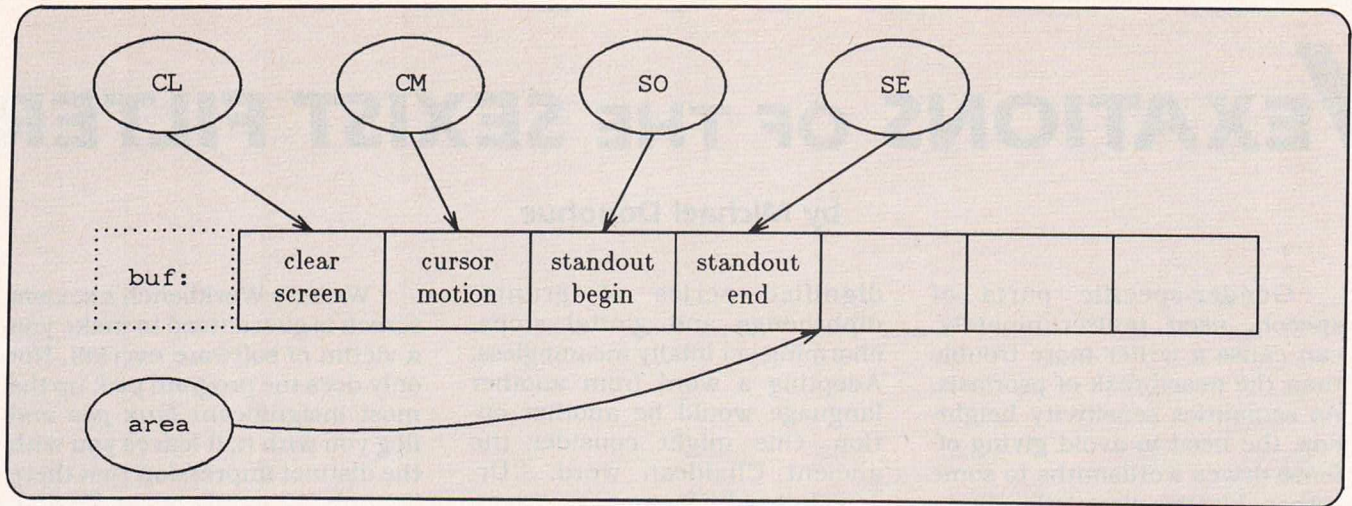


Figure 1 — After calling *tgetstr()* routine four times.

<stdio.h>, instead of a true function, we have to insulate it by placing it inside a real function. Any time a function is passed as a parameter, it is necessary to pass the address of that function. Preprocessor macros such as **putchar()** do not have an address, so they cannot be passed as a parameter to the **tputs** function.

The **main** part of the program begins by getting the terminal type from the environment, using **getenv()**. Then we call the **tgetent()** routine to fill the terminal entry buffer with data from the **termcap** database. This buffer should be 1024 characters large, because many **termcap** entries are now longer than 512. Next, we check to make sure the terminal we are using has at least 11 lines and 40 columns. If it doesn't, the program won't run properly because it uses that many lines and columns.

In this rather simple program, we only need four string capabilities — clear screen, cursor motion, standout, and standout end. So we set the area pointer to point at the beginning of buffer **buf**, and then call **tgetstr()** four times, once for each of the capabilities we need. Each time, the area pointer is advanced and **buf** is filled with null-terminated strings. The variables **CL**, **CM**, **SO** and **SE** are set to point to the appropriate string in the buffer. A diagram of string capability pointers is given in Figure 1. Pointers are represented as

ellipses, while the character array **buf** is represented in boxes, with each box being a null-terminated string. After the four calls to **tgetstr()**, **area** points to the next empty area in **buf**. Since we don't need any other string capabilities, this is the way things will remain.

Next, we clear the screen so we won't be confused by previous work on the terminal. This is done by passing the **CL** string to **tputs()**. The next three steps will be repeated three times. First, we move to a specific screen location by taking the string returned by **tgoto()** and passing it to **tputs()**. Note that column and line numbering begin with zero, so we actually begin on line 5, column 22. Second, we print a message on the screen, surrounded with terminal control sequences to put that message in reverse video. Third, we scan for an answer to our query and place the answer in the appropriate variable. After this section of the program, we should know the principal, interest rate and term of the loan.

Ultimately, we calculate the monthly payment by multiplying the principal by the fraction actually paid each month. This amount is then output to the screen, two lines further down, and labeled with text in reverse video. Finally, we move to the beginning of line 11 and exit. The only **termcap** routine we haven't used here is **tgetflag()**, because we didn't have any need for boolean capabilities. This routine

is similar to **tgetnum()**, except that the only integers it returns are zero and one.

ENHANCEMENTS

There are two things that could be improved in this program. They have not been done because the resulting program would have been too long to present here. First and foremost, the program could loop until end of file, allowing the user to change principal amount, interest rate or term when investigating the resulting monthly payment schedules. This would require turning off echoing and using "rare" mode, because numbers should be left in place if the user presses RETURN, but changed if the user enters a digit. As explained in last month's article, you attain rare mode by setting **CBREAK** on Version 7 and 4.x BSD, or by unsetting **ICANON** on System III and System V.

Less important, but still annoying, is the program's lack of robustness in the face of input error. The **scanf** statements will set variables to zero if they fail to find a valid number. This will cause a "Floating exception" because of a divide by zero in the payment calculation. The best solution would be to ignore invalid input, reprompting until a valid number comes along. Enhancements for solving these problems are left as an exercise for the reader. ■

VEXATIONS OF THE SEXIST FILTER

by Michael Donohue

Gender-specific parts of speech, used indiscriminately, can cause a writer more trouble than the heartbreak of psoriasis. As semantics sensitivity heightens, the need to avoid giving offense drives wordsmiths to some rather bizarre linguistic pirouettes.

"The chairperson has announced that one who finds himself/herself unwilling to tip the doorman or woman should leave his or her lousy car at home." In the words of *The Man (or Woman) Who Came To Dinner*, "I may vomit!"

Life might be simpler if gender-specific words were banned entirely. One could substitute a series of asterisks and let the reader fill in the blanks. Or one could invent a new word, a

dignified series of grunts, diphthongs and glottal stops, charming yet totally meaningless. Adopting a word from another language would be another option. One might consider the ancient Chaldean word, "Urzowah," which means, "man, woman, child of either sex or, in one dialect, the droppings from a large camel." One could properly advise pedestrians of either sex, "Don't step in the urzowah, urzowah."

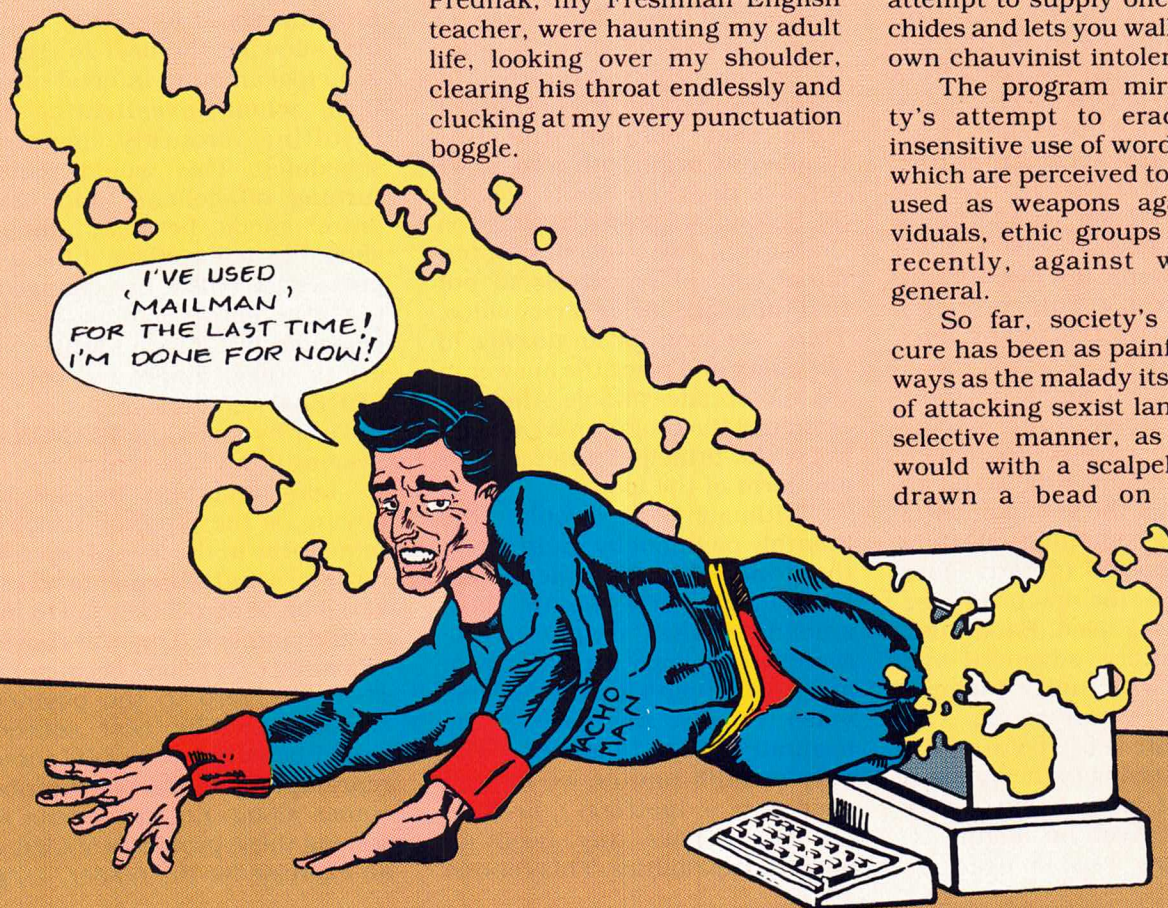
Those whose word processing includes the use of the **sexist** command included in the Writer's Workbench benefit from the nagging disapproval of a common scold that's as close as their CRT. The **sexist** command is a minor miracle as a basic grammar and punctuation checker. It is as if Mr. Prednak, my Freshman English teacher, were haunting my adult life, looking over my shoulder, clearing his throat endlessly and clucking at my every punctuation boggle.

Writer's Workbench's sexism screen is guaranteed to make you a victim of software overkill. Not only does the program pick up the most insignificant *faux pas* and flog you with it, it leaves you with the distinct impression that there are only 12 words in the English language which are not gender-specific and three of those are suspect.

A program that offers to fly to the aid of the writer's lapsed gender sensitivity cannot be all bad, and **sexist** isn't. When one falls from linguistic grace, the CRT more-or-less gently reminds that a word or phrase is gender-specific and suggests one or two insipid substitutes. Where the English language has not developed to the point that an acceptable alternative exists, **sexist** does not attempt to supply one. It merely chides and lets you wallow in your own chauvinist intolerance.

The program mirrors society's attempt to eradicate the insensitive use of words — words which are perceived to have been used as weapons against individuals, ethnic groups and, most recently, against women in general.

So far, society's attempted cure has been as painful in some ways as the malady itself. Instead of attacking sexist language in a selective manner, as a surgeon would with a scalpel, we have drawn a bead on an entire



Illustrations by Nancy Jorgensen

language and blasted away with both barrels. As a result, many have developed a verbal twitch, a semi-disabling form of stutter; sentences interlaced with "he/she" or "his/her," unfamiliar and downright uncomfortable new words on the order of "policeperson," and the painful addition of the words, "or her," as an afterthought following each masculine pronoun.

The **sexist** program prescription is no less afflictive. While some suggestions in the long-as-your-arm list of *verboten* terms and acceptable substitutes are fairly easy to swallow, others are only choked down with difficulty.

Only the most unrepentant male chauvinist would complain about substituting "equestrian" for "horseman" or "horsewoman." The latter words have always evoked visions of centaurs anyway. Still, "She was only the stableperson's daughter, but all the equestrians knew her," seems to have lost something in translation.

I suppose I will become accustomed to calling a busboy a "clearer" and giving the "attendant" the 50-cents intended for the bellboy. Also, if I am not elected to the chairpersonship of my user's group, I won't complain to my Congressperson. But, if I am not allowed to use gender-specific terms when I indicate to my host (the person

whose hospitality I am enjoying – **sexist** won't let me call her "hostess") that my spouse is the one standing across the room, surrounded by men (whom the program naively insists should be called "persons"), I am going to have to resort to pointing.

Using the **sexist** program is

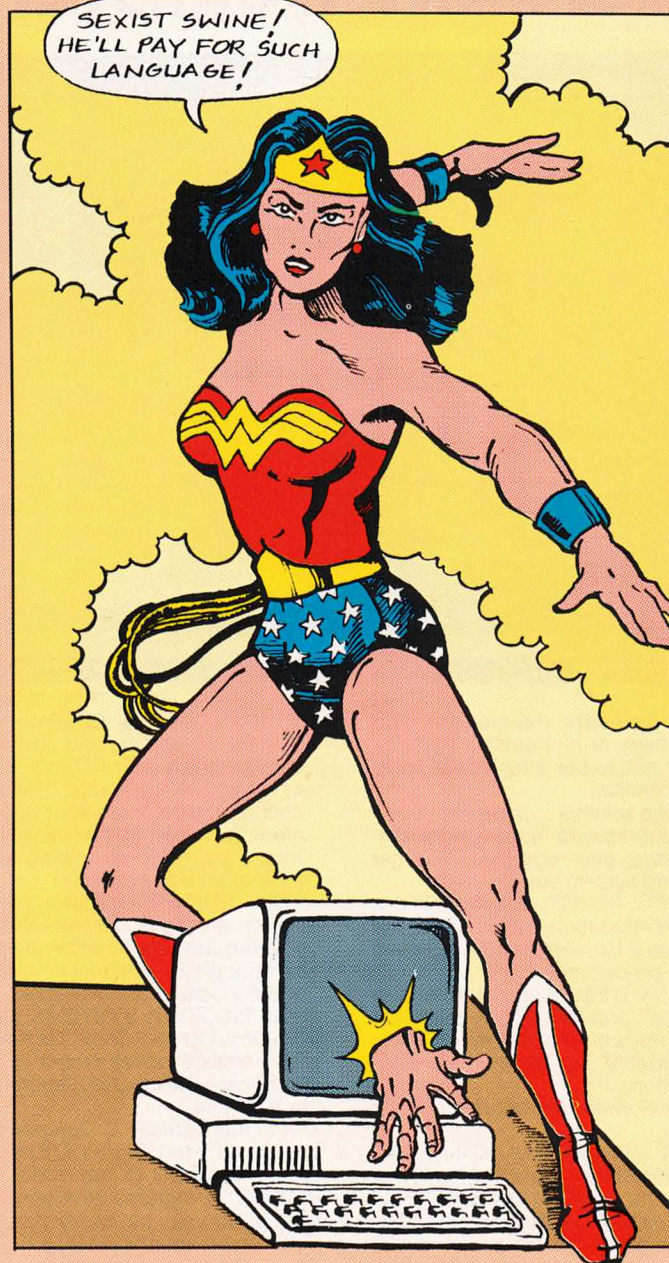
and an interminable list of potentially offending words, which, laid end to end, lead nowhere.

The average user will employ the sexist phrase dictionary once, never-no-nevermore to return. One is inundated by pages of solicitous, often useless, advice which, if followed to the letter would of necessity reduce one's writing to gibberish.

The **sexist** program's abiding virtue is that, unlike people who make it their business to point out one's use of a sexist term, it doesn't yell. This may not always be true, though. One can envision a **sexist** program of the future taking sterner measures: "Sexist swine!" the screen would fairly shout. "Your insensitivity is enough to remove varnish. Hanging is too good for you. Your kind should be horse-whipped."

The final solution to the writer's dilemma is probably not a computer program at all. Some tormented essayist is bound to propose worldwide legislation requiring that henceforth women shall be called "men" and men "women," with all attendant pronoun shifts and adjective shuffles. This would have the effect of putting jackboot on the other foot. Imagine: women (formerly "men") would raise all kinds of flak about writers' and speakers' use of the gender-

specific turn of phrase. Female (formerly "male") language awareness would be raised and there would be peace on earth at long last – a semantic utopia where the lion and the lamb shall lie down together...alongside the female (formerly "male") chauvinist pig. ■



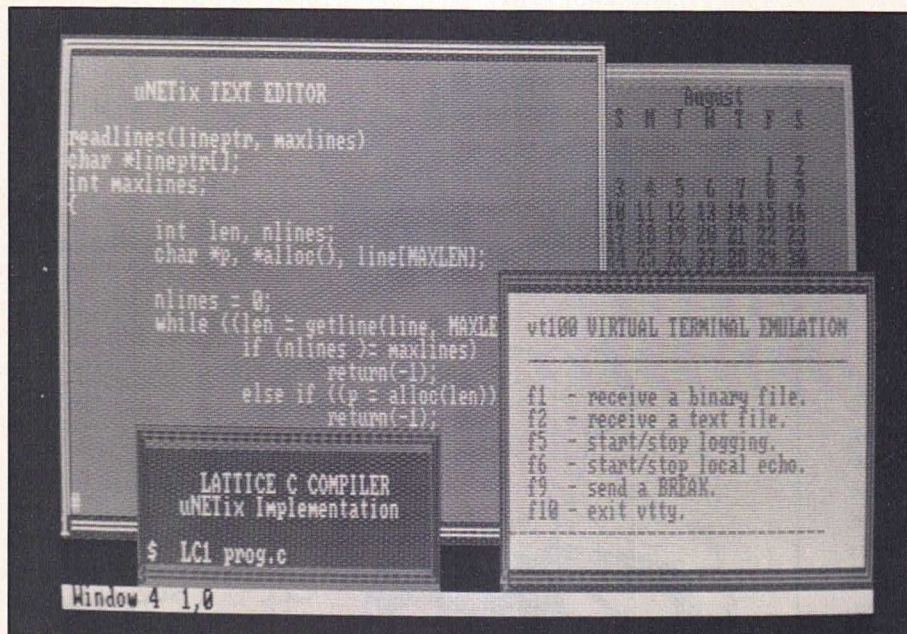
like choosing a jackhammer to do the job of a tack hammer. Not only is the writer shaken by the experience, but the work product is maimed, mangled, mutilated and thoroughly emasculated in the process. With no attempt at prioritizing its order of attack, it takes on every personal pronoun

Michael Donohue is an author and Superior Court Judge living in Spokane, Washington. He is currently at work on a novel entitled, "Uncommon Law."

Copyright 1984 Michael Donohue

SOFTWARE DEVELOPERS

LESS THAN \$400 WILL TURN YOUR PC INTO A POWERFUL DEVELOPMENT WORKSTATION.



Until now, software developers who wanted the power of high-priced development systems had to pay a high price for it. There was no choice.

Now, there's a solution... whether you're developing programs to run on minicomputers or micros, and regardless of what target operating system you're using.

If an IBM® PC, PC XT™, or compatible is part of your development environment, the uNETix Software Development Toolkit will increase your productivity.

LANTECH SYSTEMS has put an end to the "no choice" problem with a collection of powerful software products which make your PC a versatile, but low cost development workstation.

The uNETix Software Development Toolkit contains:

- **uNETix SFS** — the powerful multi-tasking, UNIX™ compatible operating system, which is at the heart of the LANTECH SYSTEMS product line. The unique "**Window Management**" feature provided with uNETix allows easy integration of applications software and permits data to be moved between various window processes (such as the Emulator or Editor described below).

- **VT100® Terminal Emulator (VTTY)** — permits up-loading and down-loading of data between your PC and a larger mini or mainframe computer. This function, of course, means that larger-scale development tasks can be run on more powerful machines, while permitting you to process other tasks locally.
- **TEXT EDITOR** — allows powerful, full-screen editing at the workstation level or you can use your familiar mainframe editor through the terminal emulator. Several popular editors are available.

The LANTECH SYSTEMS uNETix Software Development Toolkit is the solution to your software development needs, at an affordable price of \$399. Quantity discounts are available.

Optional Lattice® C Compilers are available from LANTECH SYSTEMS to develop programs locally to run under uNETix, or PC DOS™. Programs developed for uNETix can, of course, be easily ported to other UNIX environments. Lattice C Cross Compilers are also available for a variety of minicomputers.

To order your Toolkit, or to get more information contact:

LANTECH SYSTEMS INCORPORATED

9635 WENDELL ROAD, DALLAS, TEXAS 75243
(214) 340-4932 EX. 200

uNETix is a trademark of Lantech Systems, Inc. IBM is a registered trademark of International Business Machines. UNIX is a trademark of Bell Laboratories, AT&T Technologies. VT100 is a trademark of Digital Equipment Corp. Lattice is a registered trademark of Lattice, Inc. PC DOS is a trademark of International Business Machines. XT is a trademark of International Business Machines. Minimum system configuration is IBM PC with 512K RAM, monochrome or color display. RS232 port required for VT100 connection.

THE 3B2 REPORT CARD

**AT&T's
supermicro
gets put to
the test**

**by Harry Avant and
C. Andy Felong**

UNIX REVIEW's two-part assessment of AT&T's supermicro, the 3B2/300, concludes with this hands-on report on the machine's performance. Results of a suite of AIM Benchmark tests tell the objective story. Subjective remarks by systems analysts Harry Avant and Andy Felong put the figures in context. Also included are reviews of the two Teletype terminals promoted for sale with the 3B2 and an assessment of AT&T's bitmap software.

THE TERMINALS

The Teletype 5420 terminal has a 12-inch diagonal white phosphor CRT capable of displaying 80 or 132 columns and 24 lines. A 25th line shows status information. Users can choose from three types of video memory access: scroll mode, offering 78 lines of 80 columns or 51 lines of 132 columns;

window mode, capable of generating as many as four workspaces with 1584 characters each; and page mode, providing three pages of 24 lines each.

We did not care for the unit's keyboard at all. The shift key had a tendency to stick and all of the other keys required excessive force to operate. The video on the evaluation terminal was not as sharp as it should have been, given that its suggested list price is over \$1400.

The second terminal we worked with was the Teletype 5620 DMD (dot mapped display) intelligent terminal. Intelligent, in this case, is a gross understatement. The 5620 is a high resolution (800 X 1024) green display with its own 32-bit WE 32000 microprocessor! It also offers 256K bytes of RAM, 64K bytes of EPROM, two RS232 serial ports and a parallel port. One serial port is used for connection to the host CPU while the other is available for a printer. It is not clear what the parallel port is

3B2 REPORT CARD

intended for, and no mention was made of it in the documentation supplied with the unit.

Unpacking the 5620 from its shipping carton was no small feat. We should have been suspicious when we saw it took two people to lug it into the office. The monitor, excluding the keyboard, measures 18 1/2 inches tall, 13 inches wide and 17 inches deep. It weighs in at over 60 pounds! Gently sloping forward and up toward the user, the green video display measures 15 inches on the diagonal. The low profile keyboard configured to our monitor was the standard Teletype model with 83 keys divided into eight function keys across the top, a combined cursor and numeric keypad to the right, and a standard typewriter layout across the rest of the board. A three-button hemisphere mouse completed the package.

We were not fond of this keyboard either. It was similar to the 5420 keyboard in that keys had to be struck dead center to register keystrokes. A squeaking noise Teletype selected for its key click was also quite annoying after long sessions at the terminal.

In fairness, another touch typist programmer found the keyboard acceptable. We all agreed, though, that the mouse was not as nice as those used with the Apple Macintosh or Lisa. Our

hands tended to cramp after long sessions in **jim**, the DMD word processor that requires extensive use of the mouse.

Another complaint we had was monitor's slow scroll and long phosphor persistence. Coupled together, they made reading text next to impossible when scrolling.

The text layout of the 5620 terminal is 85 columns by 66 lines, which matches standard 8 1/2 by 11 paper but takes some getting used to. It should be said, though, that this format is superb for previewing a full page of text.

The functionality of the 5620 under the software supplied for the 3B2 proved to be nothing short of spectacular. Terminal firmware coupled with software running in the 3B2 enabled us to interact in multiprocess, multiwindow graphics and text modes. The key to this ability is a UNIX command named **layers** that allows the 5620's screen to be divided into as many as six windows. The windows can be tiled, overlapped or even hidden by other windows. Each may be linked to a different process in the 3B2 – though only one window can be tied to the keyboard at any one time.

Standard software for the 5620 is divided into two packages: the *Core Support Software Package* and the *Applications Development Package*. We were

only supplied with the seven programs contained in the core package, including:

layers – provides the capability of creating up to six independent virtual terminals.

321d – used to download DMD programs from a 3B2.

lens – simulates a magnifying glass capable of enlarging portions of the screen for easy viewing. Magnification ratio, viewing area and shape can be varied. When this command is active, no other 5620 resident program can execute.

jim – used to enter a mouse-based, multiwindowing word processor supporting concurrent editing of multiple files. The **jim** editor uses Smalltalk/Apple-style pop-up menus for command input. The mouse is used to select text for cut and paste operations. Since the mouse is very sensitive in **jim**, it is often difficult to point at selected parts of text. Moreover, some AT&T "user-friendly" devices seemed a bit curious to us. The "snarf" idiom for copying text to a save buffer, for instance, seemed inappropriate to a business environment – as did the **jim** coffee cup icon meaning "wait."

demo – brings up a menu of cute graphics demonstrations useful for showing off the power of the 5620.

tek4014 – allows the 5620 to emulate a Tektronix 4014 graphics terminal.

jx and **xt-driver** – used to interface the 5620 to the 3B2 computer.

We were disappointed that the *Applications Development Package* was not supplied for our review. The **proof** command is said to provide phototypesetter preview for **troff** while **cip** has

BIZPAK™ by CSSI Corp.

Full Featured Business Software For UNIX™ Based Systems

- Accounts Payable
- Accounts Receivable
- General Ledger
- Inventory
- Order Processing
- Sales Analysis
- Payroll

- * Ideal for OEM's
- * Source Licenses
- * Available Now!!

- Multi-Company
- Multi-User
- Multi-Terminal
- Multi-Warehouse
- R/M COBOL™
- Fully Integrated
- Scrolling

CSSI Corp. 10700 Lyndale Avenue South
Bloomington, MN 55420 / 612-881-4501

BIZPAK is a trademark of CSSI Corp. R/M COBOL is a trademark of Ryan McFarland Corp.

UNIX is a trademark of Bell Labs

Circle No. 240 on Inquiry Card

been described as an interactive picture editor allowing for the manipulation of arcs, circles, rectangles and text. Both commands sound useful, if indeed they live up to their billing.

On the whole, we were pleased with the 5620 hardware and software we could get our hands on. The 5620 DMD terminal utilizes the capabilities of UNIX to a high degree and offers the bitmap capability necessary to support a growing number of graphics (and text) software packages. The 5620's multiwindow concept has been proven useful by Xerox and Apple, and seems a natural addition to a multiprocessing operating system. Finally, the 5620 off-

loads intensive calculations or manipulations from the host CPU, enabling greater productivity for both system and user.

BENCHMARK RESULTS

We were supplied with the AIM Technology Benchmark Suite, courtesy of AIM, to evaluate the 3B2's performance. The tests take into consideration CPU speed, memory access time, system call efficiency, disk access time and data transfer rate. They can be run with varying mixes of disk activity, string manipulation and arithmetic processing — thus enabling users to measure how well a unit can handle specific tasks, such as: database management, word processing and scien-

tific calculations. The results reported here were produced using an equal mix of the seven possible tests. See Figure 1 for a full account. The arithmetic instruction times show that **short** and **float** operations took longer than their **long** and **double** counterparts. While this is not intuitive, it results from the true 32-bit architecture of the WE 32000 microprocessor. The 32-bit path and internal registers force the processor to work on a subset of its word, thus increasing the amount of processing needed to handle **short** and **float** arithmetic operations.

Pure test results of the sort shown in Figure 1 are not the only factors prospective buyers must

U N I X S O F T W A R E

UniPress Product UPDATE

LATTICE® C NATIVE AND CROSS COMPILERS FOR THE 8086

Outstanding software development tools

Lattice C Cross Compiler to the IBM-PC

- Highly regarded compiler producing fastest and tightest code for the 8086 family.
- Use your VAX or other UNIX machine to create standard Intel object code for your 8086 (IBM-PC)
- Full C language and standard library, compatible with Unix.
- Small, medium, compact and large address models available.
- Includes compiler, linker, librarian and disassembler.
- 8087 floating point support.
- MS-DOS 2.0 libraries included.
- Send and Receive communication package optionally available.

Hosted On

Prices: VAX/Unix and VMS	\$5000
MC68000/8086	3000
Send and Receive	500

Lattice C Native Compiler for the 8086

- Runs on the IBM-PC under MS-DOS 1.0 or 2.0.
- Produces highly optimized code
- Small, medium, compact and large address models available.
- Compiler is running on thousands of 8086 systems.

Price: \$425

COMING SOON

Amsterdam Compiler Kit

- Package of compilers, cross compilers and assemblers.
- Full C and pascal language.
- Generates code for VAX, PDP-11, MC68000, 8086 and NSC16000.
- Hosted on many Unix machines.
- Extensive optimization.

Price: Full system \$9950
Educational Institution 995

Plink — For Lattice Native

- Full function linkage editor including overlay support.

Price: \$395

OEM terms available • Much more Unix software, too! • Call or write for more information.

UniPress Software, Inc.

2025 Lincoln Highway, Edison, NJ 08817
201-985-8000 • Order Desk: 800-222-0550
(outside NJ) • Telex 709418
Mastercard and Visa

Lattice is a registered trademark of Lattice, Inc.
Unix is a trademark of Bell Laboratories. MS-DOS is a trademark of Microsoft.

Circle No. 85 on Inquiry Card

U N I X S O F T W A R E

3B2 REPORT CARD

ARITHMETIC INSTRUCTION TIMES

(microseconds per op)

	short	long	float	double
+ add	3	2	6579	3610
* multiply	13	7	15625	12500
/ divide	27	22	8929	6173

MEMORY LOOP ACCESS TIMES

(microseconds per byte)

	read	write	copy
CHAR type	7	9	11
SHORT type	4	5	6
LONG type	2	2	3

INPUT/OUTPUT RATES

(K bytes/sec)

	read	write	copy
DISK	44	21	28
PIPE			92
TTY 1		0	
TTY 1+2		0	
RAM 1-byte			94
RAM 4-byte			333

ARRAY SUBSCRIPT REFERENCES

(microseconds)

short[]	long[]
13	9

FUNCTION REFERENCES

(microsecond/ref)

0-parameters funct()	1-parameter funct(i)	2-parameters funct(i,i)
38	51	61

PROCESS FORKS

(540K bytes)
14 per second

SYSTEM KERNEL CALLS

(calls-per-second and microseconds per call)

getpid() calls:	2K calls/sec or	575 microseconds/call
sbrk(0) calls:	1K calls/sec or	843 microseconds/call
create/close calls:	92 pairs/sec or	10870 microseconds/pair
umask(0) calls:	1K calls/sec or	719 microseconds/call

Figure 1 — How the 3B2 tested out under the AIM Benchmark.

consider, however. Benchmarks alone cannot account for cost factors or questions related to support, service and expandability. Neither are such results particularly meaningful in the absence of comparative data. With that in mind, *UNIX REVIEW* will attempt to incorporate AIM Benchmark results in future hardware reviews. Over time, a useful database will presumably develop.

Even on this first pass through, we were able to make some interesting comparisons based on the AIM benchmarks measuring overall system performance. According to these standards, the abilities displayed by the 3B2 on the 36-test AIM benchmark measured out at 1351 points. Compare this figure with the results provided by AIM for three other machines subjected to the same bank of tests: a Convergent Technologies Megafame running System III scored 2436

points; and a DEC VAX 11/730 running 4.2 BSD registered 1818 points, as did a Plexus 3000 running System V. Against such competition, the 3B2's 1351-point

**One of our greatest
disappointments is
that AT&T has
unbundled the
software.**

performance lacks in luster. But, remember, these figures are relative, not absolute.

After using the 3B2 for six weeks, we are left with mixed emotions and several questions.

One of our greatest disappointments is that AT&T has unbundled the software. Many of the programs considered to be standard UNIX are either unavailable or sold only as extra options. It is surprising that AT&T has not included more Berkeley enhancements than the **vi** editor and **curses**. Certainly, the C shell would be a welcome addition.

We also question the 3B2's advertised ability to support up to 16 users. It was not possible for us to verify, but by extrapolation we would guess the real limit is somewhere around six to eight users. Of course, that figure is subject to the mix of processes being run. The norm in advertising UNIX systems is to overrate true user support, and it would appear that AT&T has followed suit.

While we cannot fault the support we received from AT&T while preparing this review, the support "real" users can expect is still

unknown. Bear in mind that we worked with a system built by AT&T's Computer Division running software developed by Bell Laboratories, using documentation prepared by Western Electric on terminals developed by Teletype. Only time will tell if AT&T will be able to coordinate these disparate groups and provide good single-point support for the 3B2 and its various related products.

It should also be remembered that the 3B2 is one of the new kids on the block. It has a long way to go before it is as well supported with software as machines based on the Motorola 68000 chip. Of particular concern are questions about when application software and languages will be available.

(Editor's Note: AT&T announced 13 software products for both the 3B2 and 3B5 on June 26.)

AT&T appears to have made provisions for an open architecture, but we cannot help but wonder when nifty third-party hardware devices will actually become available.

The 3B2 is unquestionably a well-designed computer system that should provide dependable service. Its construction is first rate and modular in design, which should provide for easy servicing. Considering its power, the computer is incredibly small. It is also capable of operating in almost any office environment. Moreover, buyers can rest assured AT&T will still be in business whenever the need for service or support arises.

Harry Avant's work at Jet Propulsion Laboratory involves the evaluation of microcomputer hardware and software. His articles have appeared in several publications concerned with microcomputers.

C. Andy Felong is a member of the technical staff at Jet Propulsion Laboratory specializing in graphics and system support. He is a long standing member of USENIX and has contributed articles to several computer-related publications. Mr. Felong also co-authored LNWBASIC, a graphics-oriented BASIC interpreter.

The authors would like to acknowledge the help they received from Mr. Tony Leone and Mr. John Chmielewski of AT&T in preparing this review.

Copyright © 1984 by Harry Avant and C. Andy Felong

U N I X S O F T W A R E

**UniPress
Product
UPDATE**

EMACS™ & MINIMACS™

Powerful Text Editing Software for MS-DOS™, UNIX™ and VMS™

EMACS

- Acclaimed Gosling Version
 - Runs under UNIX, VMS, and MS-DOS
 - Full screen multi-window editor allowing several files to be edited simultaneously.
 - Extensible and customizable via macros and built-in MLISP programming language.
 - C, Pascal, and MLISP programming assist.
 - Communicates with UNIX and VMS systems. Allows shell/DCL windows, and command execution with output directed to a window. For example, Emacs permits program compilation with source code in one window and compiler errors in another for ease of debugging.
 - EDT emulation on VMS, optional on UNIX.
 - Available on a wide range of hardware configurations including the VAX, 4.1/4.2, System III/V, Sun, Pyramid, Plexus, Cyb, Callan, Masscomp, Apple Lisa, Tandy Model 16, Pixel, Dual, Apollo, Integrated Solutions, Perkin Elmer, HP9000, NCR Tower, Cadmus, Momentum, Charles River Data, Fortune and more.
- Prices: UNIX: \$395/binary • \$995 source
VMS: \$2500/binary • \$7000/source
MS-DOS: \$375/binary • \$995/source

MINIMACS

- Faster and smaller model Gosling Emacs available for Unix.
 - Full screen editing with multiple windows.
 - Full EMACS macro capability and keybindings. Communicates with Unix via command execution with output directed to a window.
 - Minimacs brings the powerful capabilities of the finest screen editor to smaller machines.
 - Minimacs minimizes your resource load and is much smaller than the vi editor.
- Price: \$375/binary • \$795/source.

OEM terms available. • Much more Unix software, too! Call or write for more information.

UniPress Software, Inc.

2025 Lincoln Highway, Edison, NJ 08817
201-985-8000 • Order Desk: 800-222-0550 (outside NJ)
Telex 709418 • Mastercard and Visa

Emacs and Minimacs are trademarks of UniPress Software, Inc. Unix is a trademark of Bell Laboratories. VMS is a trademark of Digital Equipment Corp. MS-DOS is a trademark of Microsoft.

Circle No. 86 on Inquiry Card

U N I X S O F T W A R E

/usr/lib

Two out of three isn't bad

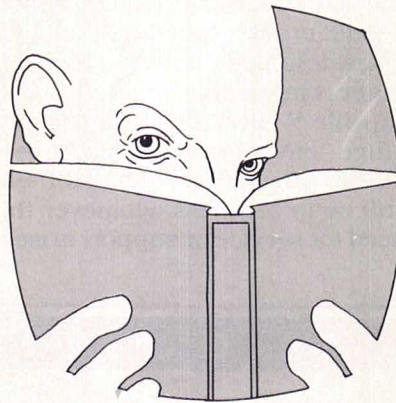
by Jim Joyce

This month, */usr/lib* reviews three publications, one on C, one on Pascal and a new edition of the *UNIX Applications Software Directory*. The Pascal book is quite good while the second edition of the directory is both larger and more complete than the first. As for the book on C...

PROGRAMMING IN C FOR THE MICROCOMPUTER USER

Robert J. Traister has given us a book on C that virtually ignores the UNIX operating system, the environment in which C is typically used. *Programming in C for the Microcomputer User* (Prentice-Hall, 1984, 190pp, \$16.95) appears as though it were written to accompany distribution of the SuperSoft C compiler. Indeed, Chapter 2 begins, "When I was assigned to write a book to introduce microcomputer users to [the] C programming language, my first task was to locate a compiler that would allow present owners of microcomputers to run programs in C on their machine." One does not normally think of book authors as being assigned. Was the assignment made by SuperSoft? One wonders.

The preface promises that "nearly every program presented in this book is a working entity unto itself." I took that to mean that the programs were running



programs until I reached page 35, where I found:

```
main( )
{
  int x;
  x = 23456;
  printf("%20d\n",x);
```

which will not compile because the right brace marking the end of **main** is missing.

But the real problem is not the missing right brace, which was possibly erased during pasteup, but the lack of indentation. Most examples in the book are flush left, with no indentations to indicate logic flow. To make matters worse, the dot-matrix examples are reduced in size to fit into the design of the book! Given that examples are important, should they not be at least as large in type size as the text?

Surprisingly, the **goto** com-

mand is presented early in the book to illustrate an endless loop:

```
main( )
{
  int x;
  QRP:
    x = 5;
    printf("%20d\n",x);
    goto QRP
}
```

This example, at least, contains indentation, but the disregard Traister shows for consistency in style is something few professional programmers would tolerate. Moreover, there is something wrong about the example. I suspect a more common idiom for an endless loop equivalent would be:

```
main( )
{
  int x;
  for(x = 5;;)
    printf("%20d\n",x);
}
```

or even:

```
main( )
{
  int x;
  for(x = 5;;printf("%20d\n",x));
}
```

Perhaps I am being a bit harsh, since Traister presents many examples of BASIC code to drive his examples in C. But I do not feel it is reasonable to

present C as if it were BASIC with semicolons. Learning C means learning to think C, just as learning BASIC means learning to think BASIC.

There is something fundamentally wrong about this book. I do not feel from reading *Programming in C for the Microcomputer User* that Traister knows C particularly well. He certainly does not know C's place in modern computing when he says in solemn earnest, "Will C become the standard programming language of today's system specialists?" The language is certainly already viewed as such. But

**It is not reasonable to present
C as if it were BASIC
with semicolons.**

Traister does not stop there. He answers the question for us: "The answers are many and varied. Some feel the answer is yes, while others say no." Note the many and varied answers.

Though the book is published by Prentice-Hall, it is not part of Brian Kernighan's series. After reading it, I can see why.

TABLE OF CONTENTS FOR PROGRAMMING IN C

1. C Programming Language (6 pp)
 2. The SuperSoft C Compiler (14 pp)
 3. Programming in C (23 pp)
 4. Input Control Programs (22 pp)
 5. Handling Character Strings (17 pp)
 6. Programs in C Language (20 pp)
 7. Going Further with C (6 pp)
- Appendices
- A. Machine Configuration (1/2 p)
 - B. SuperSoft C Compiler Standard Library Functions (53 1/2 pp)
 - C. SuperSoft C Compiler Standard Library Functions and Externals (1 p)
 - D. C Language Keywords (1/3 p)
 - E. C Language Statements (1/4 p)
 - F. Preprocessor Directives (1/4 p)
 - G. C Language Declarations (2 1/4 pp)
 - H. C Language Expressions (1 p)
- Index (4 pp)



LMC's 32-Bit Virtual Memory MegaMicro Is The-State-Of-The-Art UNIX Microcomputer

LMC's 32-bit MegaMicro provides mainframe or super-minicomputer performance at prices competitive with today's far less powerful 8- and 16-bit microcomputers. This is made possible by use of the next generation of logic chips—the National Semiconductor 16000-series. LMC MegaMicros incorporate: the NS16032 central processing unit which has true 32-bit internal logic and internal data path configured on the IEEE 796 multibus; demand-paged virtual memory implemented in hardware; and hardware 64-bit double-precision floating-point arithmetic.

The LMC MegaMicro is supplied with HCR's UNITY* which is a full implementation of UNIX** and includes the Berkeley 4.1 enhancements to take advantage of demand-paged virtual memory. Also included are C and FORTRAN. Typical multi-user systems with 33 megs. of fast (30 ms. average access time) winchester disk storage, a half meg. of RAM, virtual memory, hardware floating-point arithmetic, UNIX, C, and FORTRAN 77 are available for \$20,000 (and even less with quantity or OEM discounts).

* UNITY is a Trademark of Human Computing Resources.

**UNIX is a Trademark of Bell Laboratories.

LMC MegaMicros The Logical Alternative™

LMC

Circle No. 279 on Inquiry Card

The Logical MicroComputer Company

4200 W. Diversey, Chicago, IL 60639 (312) 282.9667



A member of The Marmon Group of companies

PASCAL UNDER UNIX

J. N. P. Hume and R. C. Holt's *Pascal Under UNIX* (Reston Publishing, 1983, 386 pp, \$22.95 hard, \$16.95 paper) is a welcome addition to books on programming. This is an introductory book more oriented toward the classroom than the professional or the hobbyist. It explains fundamental concepts well, including how to work with UNIX.

Although there are fragments of code, there are also enough complete programs to provide a context for learning the language. Furthermore, the programs mix discussions of the language with words of wisdom about the act of programming itself. Chapter 10 is a good, clear discussion of problem solving and, appropriately enough, is entitled "Structuring Your Attack on the Problem."

Although Hume and Holt explicitly name Bill Joy's Pascal compiler, they do not limit their discussion to it. The compiler's diagnostics are wonderfully helpful to beginners, but the authors even-handedly discuss systematic ways of exploring

syntax errors without making much specific reference to it.

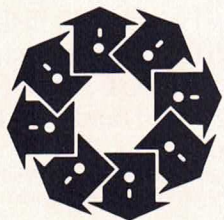
Hume and Holt present Pascal as a series of language subsets that develop the reader's awareness of what a programming language can do. Although their chapter on pointers is disappointingly

**I wish this book had been around
when I was teaching Pascal.**

short, they do treat the subject well. My experience is that the topic of pointers tends to be the single hurdle that stops Pascal (and C) programmers in their tracks.

In summary, I wish this book had been around

**UNPARALLELED
PERFORMANCE
and PORTABILITY
in an ISAM PACKAGE
at an UNBEATABLE
PRICE**



c-tree™
BY FAIRCOM

2606 Johnson Drive
Columbia MO 65203

The company that introduced micros to B-Trees in 1979 and created ACCESS MANAGER™ for Digital Research, now redefines the market for high performance, B-Tree based file handlers. With c-tree™ you get:

- complete C source code written to K & R standards of portability
- high level, multi-key ISAM routines and low level B-Tree functions
- routines that work with single-user and network systems
- no royalties on application programs

\$395 COMPLETE

Specify format:

8" CP/M® 5¼" PC-DOS 8" RT-II

for VISA, MC or COD orders, call toll free
1-800-232-3344

Access Manager and CP/M are trademarks of Digital Research, Inc.
c-tree is a trademark of FairCom.

© 1984 FairCom

when I was teaching Pascal. It is well written and introduces advanced concepts in a matter-of-fact, reader-friendly manner.

TABLE OF CONTENTS FOR PASCAL UNDER UNIX

1. Introduction to Structured Programming (7 pp)
2. The Computer (10 pp)
3. PS/1: Programs that Calculate and Output (13 pp)
4. Putting the Program on a Computer (12 pp)
5. PS/2: Variables, Constants, and Assignments (20 pp)
6. PS/3: Control Flow (24 pp)
7. Structuring Control Flow (14 pp)
8. PS/4: Arrays (11 pp)
9. PS/5: Alphabetic Information Handling (24 pp)
10. Structuring Your Attack on the Problem (13 pp)
11. The Computer Can Read English (9 pp)
12. PS/6: Subprograms (16 pp)
13. Searching and Sorting (11 pp)
14. Making Sure the Program Works (10 pp)
15. PS/7: Files and Records (15 pp)
16. Data Structures (21 pp)
17. PS/8: Pointers and File Buffers (9 pp)
18. Scientific Calculations (15 pp)
19. Numerical Methods (17 pp)
20. Programming in Other Languages (11 pp)
21. Assembly Language and Machine Language (14 pp)
22. Programming Language Compilers (25 pp)
- Appendices
 1. Specifications for the PS/K Language (22 pp)
 2. The Syntax of PS/K (4 pp)
 3. Predeclared Pascal Functions (2 pp)
 4. Summary of Pascal Input/Output Features (3 pp)
 5. Collating Sequence (4 pp)
 6. Syntax Diagrams for Standard Pascal (4 pp)
- Index (12 pp)

UNIX APPLICATIONS SOFTWARE DIRECTORY, 2ND EDITION

It seems that UNIX software is exploding all around, and Ray Jones of Onager Publishing has been forced to a second edition of his directory in less than a year! Up from 120 pages, the 198-page guide is perfect-bound and typeset. The price has gone up to \$50 because of the increased cost of printing a larger volume, Jones says.

Users are warned that the material has been compiled but not tested individually. Still, the company name and address list in Section 5 puts users

Continued to Page 93

Unix, C & You . . .

Technologies such as UNIX and C provide a valuable resource for meeting data processing needs. But, your most valuable investment lies in your knowledge of how to use these tools effectively and efficiently.

At NCR our business is providing you with the technology to satisfy your information needs. At NCR's Customer and Support Education division our business is providing you with the resource to harness that technology - YOUR KNOWLEDGE. In the past 10 years alone we have trained over 100,000 students in every aspect of data processing and completed over half-a-million days of instruction.

To the UNIX and C user we provide a full curriculum of hands-on training designed to meet the needs of Programmers, System Administrators, Operators and Managers.

	UNIX Operating System	UNIX System Administration	C Programming
Length	5 days	3 days	5 days
Tuition	\$950	\$570	\$950
Chicago	Aug 20 Oct 8 Dec 3	Oct 15	Sept 24 Nov 5
Dallas	Sept 10 Dec 3	Sept 17 Dec 10	Nov 12
Dayton	Aug 13 Sept 10 Dec 3	Sept 24 Oct 29 Dec 17	Aug 20 Sept 17
Los Angeles	Oct 8	Oct 15	
New York	Oct 1 Nov 26		Sept 10 Nov 5 Dec 10
Houston	Oct 22	Oct 29	

ENROLL TOLL-FREE

(or call to receive a UNIX information packet)

1-800-845-CASE

(Ohio 1-800-841-CASE)



Circle No. 88 on Inquiry Card



Confirmation of enrollment will be made upon receipt of purchase order or check for full tuition, refundable up to seven calendar days prior to start of class.

UNIX is a Trademark of Bell Laboratories.

. . . together

RULES OF THE GAME

I've got a secret

by Glenn Groenewold

We usually think of a secret as information known only to one person or a handful of people. It can be difficult to accept the idea that there can be such a thing as a secret that hundreds, perhaps thousands of people know. But, in the eyes of the law, a trade secret is not defined by the number of people who might be in on it.

Most people are apt to think of a trade secret as a recipe or formula for a product such as a sauce or beverage. How often are we told to purchase some item because of its "secret" ingredient? A trade secret is by no means limited to such things, however. A customer or supplier list could also be a trade secret – and an exceedingly valuable one at that. So, too, can methods for doing things or producing products.

Trade secrets have been around for a long time and constitute one of the most important categories of legally protected intellectual property. Some lawyers argue that it is the *most* important category. Yet it's easy for people who are not lawyers to overlook trade secrets and concentrate instead on copyrights and patents.

Maybe this happens because there isn't anything to *do* to create a trade secret – except keep it a secret. It's not necessary to register or file anything. Another possible reason for trade secret protection being overlooked is that



it is derived from state laws rather than from federal sources. (Be aware, though, that even though trade secret protection comes entirely from state laws, suits which involve them may take place in federal courts.)

In the computer world, trade secrets are relied upon to protect both marketing information and products such as computer programs. You can't copyright a list of customers or hot prospects because it doesn't express an idea, but it's still worth much to you to keep your competitor from using it. Thus, you keep the list as your trade secret.

This means you guard it. You don't leave it lying around where anybody who wanders in can see it. And most emphatically, you don't give it out to just anyone.

But Joe, your trusted sales manager, knows what names are on the list since he couldn't do his job otherwise. Let's say that Goliath Corporation makes Joe an offer he can't refuse, and off he goes to join its staff. Can Joe now reveal the contents of your list to Goliath?

Not if you've done a proper job of maintaining the list as your trade secret. This means, among other things, that you took the trouble to have Joe sign an agreement in which he acknowledged that any information he might learn about your operation while working for you would remain your trade secret which he agreed not to disclose.

PROTECTING INTELLECTUAL PROPERTY

What if Joe sticks to his agreement and keeps the information to himself, but uses it to go out and solicit business from one of your best prospects? Unfortunately the answer has to be, "it depends." We don't have the space to go into all the contingencies here but the main point to remember is that Joe can't pass your information on to anyone else, provided you've done your job in maintaining its status as a trade secret.

Aside from their use in marketing, trade secrets are highly important as a means of

protecting intellectual property both before and after product development. Computer programs are commonly protected by claiming trade secret status for them, often in combination with other types of protection. It is possible, for example, to copyright software and yet still retain its trade secret status.

In order to understand the circumstances under which this can be done, we have to go back to the basic premise that a trade secret is just that — a *secret*. If something has been made available to the entire world, it can't be a trade secret, no matter how often you jump up and down and insist that it is. Therefore, a computer program which has been put on the market to be sold outright is not entitled to trade secret protection (it can and should be copyrighted, however).

LICENSING

But if instead of selling the program you *license* it for use by someone, retaining the ownership yourself, what then?

If it's a true licensing situation and you've required your licensee to sign an agreement to keep your program confidential, it's pretty certain the courts will consider the program to be your trade secret. (This, of course, is exactly the manner in which UNIX is licensed for use.)

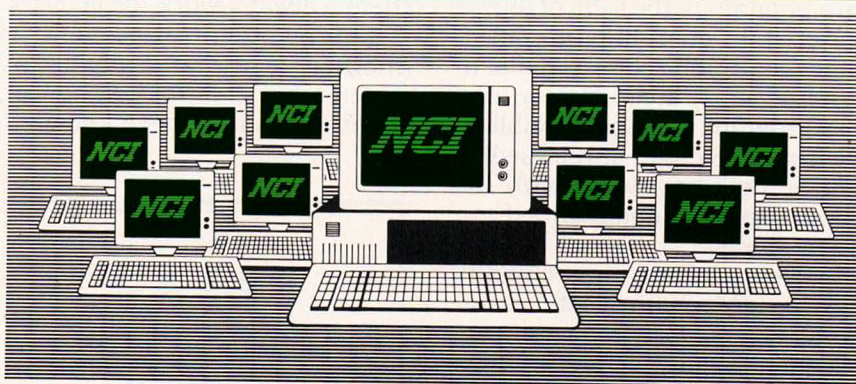
On the other hand, some "licenses" in the software distribution field are really outright sales, with neither party to the transaction having any interest in what the other does once the money and software have changed hands. The courts have little difficulty in seeing these transactions for what they are — as sales.

This brings us to another very important point regarding trade secrets. In general, once you permit public access to your trade

secret, you have lost it for good. You may have carefully and properly licensed your program to 99 users, but if you slip up and sell it

on the market to the hundredth, that program is no longer a trade secret. That also means that trade secret restrictions no longer apply

A new breed in speed **NCI COHERENT**



The only UNIX compatible operating system that supports 11 simultaneous users on 1 IBM PC XT

Only NCI COHERENT offers powerful multiuser capability with UNIX compatibility.

NCI made COHERENT a multipurpose operating system. We extended it to suit a wide variety of applications. For example, it can now be used to turn a PC into a data multiplexer or an inexpensive database machine in a local area network. NCI COHERENT can also be used for high speed protocol conversion and process control. We've even written a real-time version of COHERENT, exclusive to NCI, which will run on an intelligent peripheral board.

NCI has enhanced the COHERENT kernel making it much faster. Then we added commands that make programming easier, including

a source code control system, plus many UNIX System V features. We also gave it exclusive support for a wide variety of hardware and PC compatibles. NCI COHERENT comes with full technical support and complete documentation, organized in the UNIX manner.

A variety of licensing options are available including runtimes, production runtimes, and driver kits.

Remember, when you license COHERENT your license includes multiuser and our low license fees

make your application far more economically feasible.

The NCI technical design team that engineered the improvements to COHERENT is also available on a contract basis.



Performance engineered into every product.

NETWORK CONSULTING INC.
Suite 110, 3700 Gilmore Way,
Burnaby, B.C. Canada V5G 4M1
Phone: (604) 430-3466

*UNIX is a trademark of Bell Laboratories. COHERENT is a trademark of Mark Williams Co. IBM PC and IBM PC XT are trademarks of International Business Machines Corp.

to your first 99 licensees. The situation is different with copyrights, where the law does forgive occasional goofs.

Trade secrets can complement other means of protecting intellectual property. For instance, it's possible to register a copyright on a program in the form of source code while guarding the object code as a trade secret. This is in fact frequently done. Also, so long as a program remains unpublished (that is to say, unreleased) it can be both registered with the Copyright Office and guarded as a trade secret.

Even patents are not totally incompatible with trade secret protection. As long as a patent application remains pending, which may very well be for several years, the subject of the patent can continue to be a trade secret. Modifications and refinements made in a process or program upon which a patent has been issued can also be guarded as trade secrets.

In contrast with patents and copyrights, there is no time limit on the life of a trade secret. This can be very useful, and some trade secrets have become truly venerable.

TRADE SECRET CAVEATS

But the huge disadvantage of trade secrets is that if your secret ever gets out, it's gone. Period. This is true even when you've done nothing to disclose the secret but *someone else has succeeded in learning it through legitimate means*.

What is meant by "legitimate means?"

First off, there's reverse engineering. If simple analysis discloses the secret ingredients in a cocktail sauce, that's that. Anybody can make exactly the same sauce, though they'll have to market it under a different trade name.

However, there are limits. If the reverse engineering process requires an effort the magnitude of the Manhattan Project, the courts will consider it dirty pool, not legitimate means. Exactly where the line will be drawn is a bit fuzzy, which is one of the reasons there's always work for lawyers.

Independent discovery is another legitimate means by which a trade secret can be lost. Great minds have been known to

**There is no
time limit on
the life of a
trade secret.**

flow in similar channels, and when they do, the first possessor of a secret is out of luck. To use a mundane example, Goliath Corporation can develop its own list of prospects by plodding through and contacting all of the people who could conceivably be potential customers. If Goliath's list turns out to look just like yours, well, that's the way the ball bounces.

We all have some idea of what might be an *illegitimate* means of obtaining a trade secret. We've already mentioned that Goliath can't hire Joe away from you with the idea of getting your list from him. Neither can it legitimately obtain the list through espionage. If, however, Goliath gets the list innocently, say as the result of one of your disgruntled ex-employees volunteering it without identifying it as yours, the law is not so clear.

It should be evident by now that secrecy is everything in main-

taining trade secrets. Does this mean you have to build an electrified fence around your operation and take on the support of a pack of guard dogs?

Fortunately not. It's important that security measures be tailored to the circumstances; you're guarding intellectual property, not Fort Knox.

Start with the idea that if there's no reason certain people should have the secret, it shouldn't be volunteered to them. At Coca-Cola, it's rumored that no single individual alive knows the complete formula for the famous beverage. While this approach probably wouldn't apply to your situation, try to keep the number of people who know *everything* to a minimum. Make certain that each employee signs a non-disclosure agreement, and that every licensee signs an agreement to keep your information confidential. If sign in/out logs are used, make sure they're scrupulously maintained.

But keep in mind that too much security can be counter-productive. If you devise a system for tracking documents that is so cumbersome that no one will use it, you may be better off doing nothing. This failure to observe required security procedures can be used as evidence against you in court. It is proof that the secret which was supposed to be protected has been abandoned. At this point, it's bye-bye trade secret. And that, all of us but Goliath will agree, would be a great pity.

Copyright © 1984 by Glenn Groenewold

Glenn Groenewold is a California attorney who divides his time between computer law and service as an administrative law judge. He has been active in trial and appellate work and has argued cases before the state Supreme Court. ■

Tricks

Continued from Page 29

produce a list of all the lines in a file following **.SH** commands:

```
/.SH/ {  
    n  
    p  
}
```

The part of script enclosed between slashes finds all lines that begin with a **.SH** command. The caret (^) specifies that the command be found only at the beginning of a line. Each time **sed** finds a line that matches this specification, it executes the two commands between the braces. The **n** (next) causes **sed** to read the next line of text and the **p** (print) causes **sed** to display the line it is working with. The effect is that **sed** prints all lines following **.SH** commands.

If you store this script in a file named "s1", any one of the following commands will cause **sed** to process a file named "textfile." The **-n** option makes **sed**'s only output lines as specified by the **p** command; without this option, **sed** outputs all lines of the file in addition to those specified by **p**. The **-f** option causes **sed** to take its instructions from file (s1) instead of the command line:

```
sed -n -f s1 textfile  
sed -n -f s1 textfile | more  
sed -n -f s1 textfile > heads
```

The first two commands will list the heads on your terminal while the third will put the list in

**With vi and nroff, there is always
more than one way to perform
any task.**

a file named "heads." Refer to the earlier discussion of **more** for information on the **more** utility.

Listing mm Heads. If you are marking headlines with a macro on the same line, the task is even simpler. You can use **grep** to find all the heads with the following command:

```
grep '^\.H' textfile
```

This command displays all lines in **textfile** that begin with a **.H** command. As with the

previous **sed** command, you can pipe the output of this command to **more** or redirect it to a file.

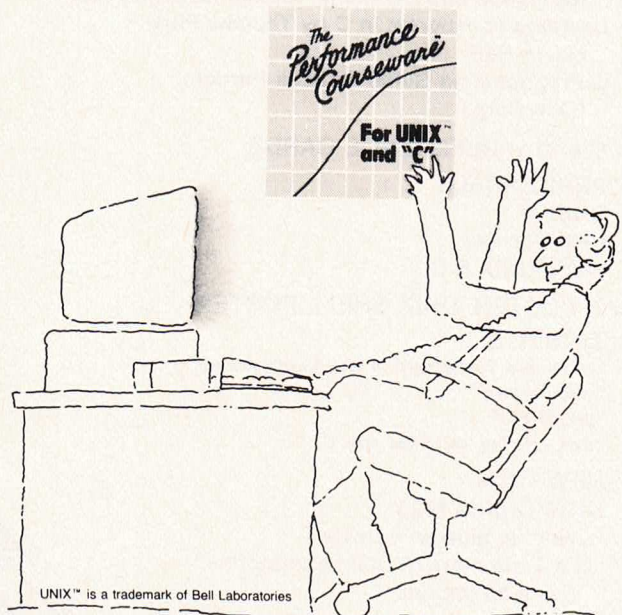
You can learn to use complex programs such as **vi** and **nroff** on a variety of levels.

The first level of learning involves understanding how to use a few basic commands and produce output. Many people never get beyond this basic level – though they learn to use the tools very effectively.

But with **vi** and **nroff**, there is always more than one way to perform any task. Some are quicker and more elegant than others. This article has presented some ideas for using UNIX tools to make your writing/editing/production job go more smoothly. If you frequently use UNIX for document production, you may want to expand some of the ideas.

Mark G. Sobell is the author of "A Practical Guide to the UNIX System" (Benjamin/Cummings, 1984). His 10 years in the computer industry include programming and technical writing experience. Mr. Sobell has been involved in UNIX for four years and is currently a consultant in the San Francisco Bay area. ■

Great Learning Exhilarates



And it saves you money.

CALL NOW USER TRAINING CORPORATION
(408) 370-9710
591 W. Hamilton Ave. • Campbell, CA 95008

Circle No. 264 on Inquiry Card

Workbench

Continued from Page 56

though, you gain the ability of instructing the system to carry out general policies, repetitive instructions and variations based on tests and calculations. In expert hands,

the formatters can produce accurate, consistent and attractive output on projects of enormous complexity and scope.

The **nroff** (nprinter output) and

troff (typesetter output) allow you enormous power and flexibility in designing your output – but programming them is somewhat akin to using assembly language. Rather than working conceptually, you have to specify exactly what you want in great detail, using terse commands whose operation is often better explained historically than logically.

The macro packages, unfortunately, are only a small step towards a high-level descriptive language. Like macro instructions in assembly language, each text macro replaces groups of up to hundreds of formatting commands. Non-programmers may also find them frighteningly complex, however, so they don't represent the ultimate in UNIX text processing. In syntax and format, they read like telegraphic abbreviations or shorthand code.

The **mm** ("memorandum macro") package occupies the central role in the Documenter's Workbench. This package is delivered in a version for use with **nroff** and **troff**, plus another for use with the speedier but simpler **sroff** printer-style formatter.

The **man** macro set is more specialized. It's customized for producing manuals in general, with defaults set for the actual manual style used for the Documenter's Workbench documentation.

For making viewgraphs and slides (including overhead transparencies), the workbench includes the **mv** macro package. Although obviously developed for a specific Bell Labs installation, this group of macros provides a general language to describe the hierarchical relationship between text blocks for inclusion in a single frame. Working from the slide dimensions and the desired text, the macro package then calculates

THE INDEPENDENT UNIX® BOOKSTORE

OVER 60 UNIX AND C ITEMS IN STOCK INCLUDING

■ UNIX BOOKS

- The UNIX System by Stephen R. Bourne (Addison-Wesley)
- Operating System Design: The XINU Approach by Douglas Comer (Prentice-Hall)
- The UNIX Programming Environment by Brian Kernighan and Rob Pike (Prentice-Hall)
- A Practical Guide to the UNIX System by Mark G. Sobell (Benjamin/Cummings)

■ C BOOKS

- The C Puzzle Book by Alan R. Feuer (Prentice-Hall)
- The C Programming Language by Brian W. Kernighan and Dennis M. Ritchie (Prentice-Hall)
- Learning to Program in C by Thomas Plum (Plum-Hall)
- C Programming Guide by Jack Purdum (Que Corp.)

■ C and vi REFERENCE CARDS

■ PERIODICALS

- UNIQUE
- UNIX Review
- World UNIX & C

■ vi POSTER, UNIX SHELL POSTER

■ T-SHIRTS

- "UNIX is a Trademark of Bell Laboratories"
- "-rwxrwxrwx"
- "grep for it"
- "awk: bailing out near line 1"

■ NEW ITEM

- UNIX Coffee Mug
- "UNIX" in blue on outside
- "is a Trademark of Bell Laboratories" inside lip imprint
- Porcelain mug, kiln-fired permanent color

Call or write for a complete catalog

Mail and phone orders only.

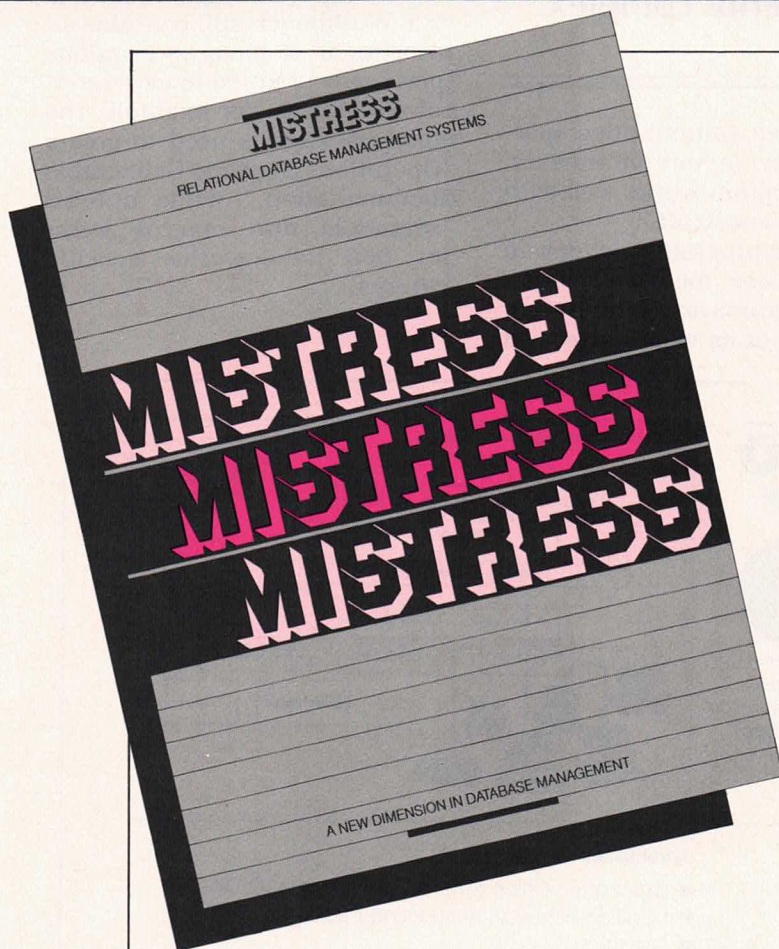
We ship anywhere in the U.S.

Contact us for shipping elsewhere.

* UNIX is a Trademark of Bell Laboratories



A NEW DIMENSION IN DATABASE MANAGEMENT



**Come meet
MISTRESS
at Comdex (Booth S9424-26)
and Uniforum (Booth 703)**

MISTRESS is the fully relational database management system (RDBMS) for UNIX.* It features the Structured Query Language (SQL*) for the end user as well as standard programming interfaces to the C language for the DP professional. Advanced concepts include variable-length character fields, dynamic storage allocation, and B+ Tree indexing. **MISTRESS** has been designed exclusively for the UNIX environment and is totally written in C.

MISTRESS/32 is the advanced relational database management system for extended addressing UNIX products. **MISTRESS/32** features enhanced capabilities for security, recovery and data integrity, as well as a fully integrated report writer and screen interface. **MISTRESS/32** is the recommended system for more demanding applications.

*UNIX is a trademark of Bell Labs. IBM and SQL are trademarks of International Business Machines.

RHODNIUS Incorporated

10 St. Mary Street, Toronto, Ontario, Canada M4Y 1P9

(416) 922-1743

Telex: 06-986766 TOR.

the type size and spacing that will make for the best fit.

After macro packages, preprocessors represent the next step up in the workbench. Although these utilities still require a close reading of the manual, a good memory and a willingness to describe the desired format in a restricted and precise way, they're sufficiently English-like to be used by non-experts.

For papers or publications concerned with mathematics, the **eqn** and **neqn** preprocessors (for use with **troff** and **nroff**, respectively) construct the formatting commands needed to include equations and math symbols in text. Because the special symbols needed for displaying equations are not available on most printers, **eqn** produces best results when paired with **troff**.

Tables occur frequently in technical material, so the **tbl** preprocessor seems quite popular with users. This utility lets you

Non-programmers may also find them frighteningly complex.

specify boxes, column titles, spacing and other layout parameters. The **tbl** program works well with both **nroff** and **troff**.

The **pic** program provides an easy language for constructing simple diagrams and illustrations. Although not as widely used yet

as some of the other preprocessors, this may eventually become one of the more popular programs in the Documenter's Workbench. Only typesetter output is currently supported, but as laser and other electrostatic printers blur the distinction between such output classifications, a future edition of **pic** should support other graphic output devices as well.

Strangely enough, Documenter's Workbench still contains an assortment of tools and options specifically oriented toward internal AT&T use. For example, the **mosd** package is used to create "Operations Systems Deliverable Documentation," while **non-btl** "reinstalls **mm** macros without Bell Laboratories specific features."

FASTEST UNIX MICRO THE X-286

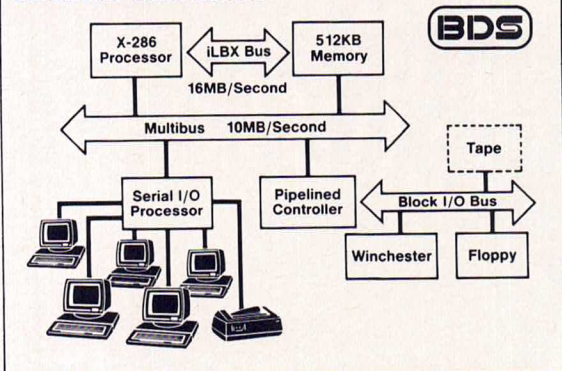
The X-286* Supermicrocomputer is the highest performance UNIX* micro available today. Only with the advent of Intel Corporation's iAPX 286 microprocessor has it become possible to meet the need for a high-performance multi-user computer with a microprocessor-based system.

System Specifications

- 80286 Microprocessor with On-chip Pipelining and Memory Protection
- 80287 Numeric Co-processor (80-bit registers)
- 512KB Error Detecting and Correcting RAM (expandable to 16MB)
- iLBX Bus for High-speed Nonarbitrated Memory Access
- IEEE 796 (Multibus*) System Bus
- Intelligent Communications Controller
- 65 or 144 Megabytes of Winchester Disk (30ms average access time) (expandable to over 1 gigabyte)
- XENIX* 286 Operating System
- 6 users (expandable to 16)

*UNIX is a trademark of Bell Laboratories. Multibus is a trademark of Intel Corporation. XENIX is a trademark of Microsoft Corporation. X-286 is a trademark of BDS, Incorporated.

DISTRIBUTED ARCHITECTURE



Available Software

- Quadratron Office Automation Software (Q-Office)
- Certified Business Accounting Packages
- Graphics
- Data Base Management
- TouchStone MS/DOS to UNIX Communications Package
- Languages

Worldwide Support and Service

Available NOW

To learn more about the high performance of the X-286, contact BDS, Incorporated, 1400 Shepard Drive, Sterling, Virginia 22170, (703) 430-0800.

/usr/lib

Continued from Page 85

in touch with a vendor's representative in most cases.

Thoughtfully included as Section 6 is a Vendor/Order Information form, so anyone who has a

run on UNIX systems will want this second edition. There is a note on the last page indicating that a hardware directory is in the works. I will let you know when that comes out.

TABLE OF CONTENTS FOR THE UNIX APPLICATIONS DIRECTORY, 2ND EDITION

1. Foreword (5 pp)
2. Cross Reference Matrix (12 pp)
3. Detail Description of Operating System Software (11 pp)
4. Detail Description of Applications Software (149 pp)
5. Company List (19 pp)
6. Software Data Entry Form (2 pp)

Jim Joyce is President of International Technical Seminars, Inc., a firm committed to UNIX training, and founder of the Independent UNIX Bookstore. For answers to your questions about books, call 415/621-1593.

**Ray Jones of Onager Publishing
has been forced to a second
edition of his directory in less
than a year!**

software product that did not make it into the second edition can submit for inclusion in the third.

Anyone wanting to locate software packages that

ACUITY® business software is compatible with any budget, and all these systems:

UNIX-based Micros	VAX, VMS or UNIX
PRIME	Convergent
IBM-PC	Harris

With prices from \$700 to \$6500 for a fully supported package, any size company can afford our general accounting and specialized project cost software.

Packages available include project management, labor/ODC forecasting, work breakdown structure, customer order processing, bill of materials processing and inventory management.

Plus a complete set of accounting software including general ledger, payables, receivables, payroll and fixed assets.

Call (619) 474-2010 for details.



**COMPUTER
COGNITION**

225 West 30th Street, National City, California 92050

Circle No. 89 on Inquiry Card

Now Affordable Power and Performance

Get all the power you need and all the performance you want from a computer system unmatched in its class — at a realistically affordable price!

The **M-5** combines the processing speed, large instruction space and sophisticated instruction set of the Motorola 68010 processor with the wide variety of peripherals available and fully supported under UNIX* 4.2BSD. The **M-5** system features 1.22 MIPS performance, 32 bit internal addressing, and allows you to expand the high-speed memory to 4 megabytes of dual-port no wait state memory.

Basic configuration of the M-5 system is: 68010 CPU, 2 1/4 MB memory, 10 user, 80 MB removable disk drive, 75 ips, 1600 bpi tape drive, and UNIX 4.2BSD.



We've saved the best for last. Price for the **M-5** computer system is \$29,885. When you compare power and price, you can't afford not to have an **M-5** system.

* UNIX is a registered trademark of Bell Laboratories

Circle No. 90 on Inquiry Card



Medical Informatics, Inc.

12023 Lakeshore Blvd. • Cleveland, OH 44108
216-249-8353

OFF THE SHELF

Hooking into the portable computer market

by Doug Merritt

As much as we all know and love the way UNIX works when it's driven by a big engine, the sheer number of personal computers out in the market make them a force to be reckoned with. With this in mind, I present the following review of two communications packages targeted at making IBM PCs work with UNIX systems.

Communique by COSI (Ann Arbor, MI) and *PCworks* by TouchStone (Seal Beach, CA) are two packages that allow PCs to be used as UNIX workstations. There are a number of similarities between the two, which I have listed below:

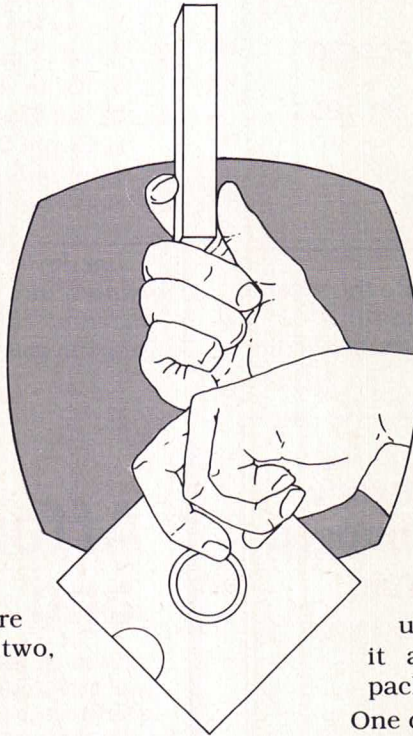
- terminal emulation
- file transfer from UNIX to PC
- file transfer from PC to UNIX
- optional automatic login
- use of standard RS232

TouchStone suggests that *PCworks* be thought of as "an inexpensive alternative to a local area network," since files can be exchanged between two or more PCs hooked up to the same UNIX system.

COSI looks at *Communique* as a tool to be used for creating an integrated environment in which the operation of PC and UNIX software might freely mix, transparent to the user.

PCworks has a more polished and elegant user interface that uses some PC graphics characters in drawing its menus. It is also simpler to understand and is thus well suited for straightforward end user applications.

Communique, on the other hand, is more flexible and dovetails nicely with COSI's *Visual Menu*



system, a specification-driven tool for creating menus. This implies that a systems house or sophisticated user could use *Communique* together with *Visual Menu* to create highly integrated user packages. Without *Visual Menu*, *Communique* is actually quite similar to *PCworks*, though it is somewhat less intuitive in operation for first time users. *Communique* is marginally more powerful than its counterpart, however. For these reasons, *Com-*

munique would probably be of most use to systems houses prepared to use it as a tool for creating integrated packages.

One definite advantage that *PCworks* provides to end users is its ability to upload all necessary software from the PC to the UNIX system under menu control. Thus, much less expertise is required to bring up *PCworks*.

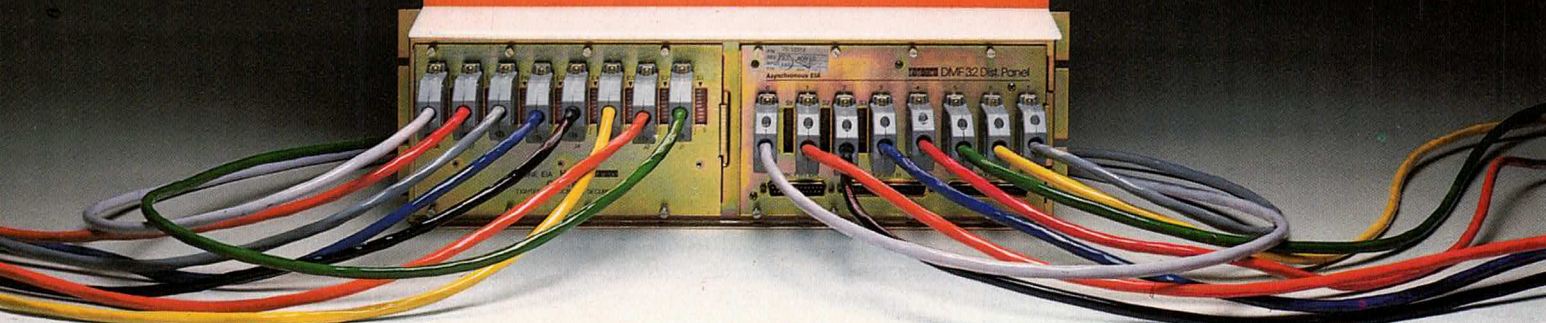
Another advantage of *PCworks* is that it has been ported to a fairly wide variety of systems, including the AT&T 3B2, DEC VAX and PDP 11, Dual 83 series, IBM PC-XT, and Zilog and Altos systems.

By comparison, *Communique* supports a much smaller number of systems, notably the VAX, although COSI has expressed an interest in porting *Communique* to other systems on request from OEMs and manufacturers.

Communique, while not self-bootstrapping like *PCworks*, is nonetheless fairly simple to bring up. There are six UNIX utilities that COSI has written to provide all UNIX functions necessary to *Communique*. Installation consists of simply copying these utilities to a UNIX directory.

For your compatibility connection call us

First



■ Pyramid 90x ■ Tower 1632 ■ CADMUS 9000 ■ Spirit/23 ■ Spirit/73 ■ PDP-11/23+ ■ PDP-11/73 ■ MicroVAX

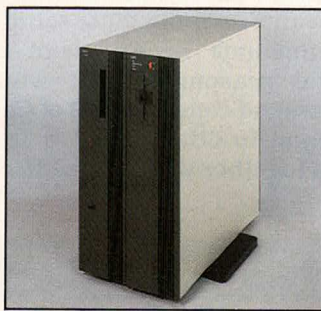
Compatibility — today's by-word for the computer industry. Compatibility between computer types, peripherals, protocols and operating systems. Compatibility that shows up in the bottom line with increased through-put, productivity and efficiency.

First Computer Corporation is *your* compatibility connection. Over seven years of technical experience in systems integration lets us answer your questions about compatibility. Our superior product portfolio allows us to select the right peripherals, components, terminals, disk or tape drives and software compatible to your needs. And our availability and service on all your product needs is second to none.

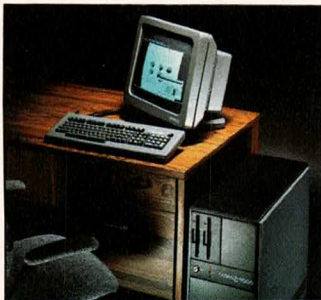
UNIX

Number 1 for compatibility

Today's state-of-the-art UNIX* operating system offers you the ultimate in compatibility: 1) networking ease — including Ethernet**, DECnet† and fiber optic links 2) system selection independence 3) greater portability 4) upward expandability and flexibility for future needs 5) heightened productivity.



NCR's Tower 1632 — a UNIX based machine that delivers the power and performance of a mini at the size and price of a micro.



The CADMUS 9000 offers high resolution graphics, windowing capabilities and a high performance UNIX based operating system.

First Computer Corporation knows UNIX. We offer a wide variety of UNIX-based systems: NCR's Tower 1632, CADMUS 9000, Pyramid 90x, PDP-11 and VAX-11.

So, when you're looking for your compatibility connection, the increased efficiencies of UNIX and you want it all from one source, call us First. You'll receive expert knowledge, superior product selection, reliable service and prompt delivery.

* UNIX is a trademark of Bell Laboratories.
** Ethernet is a registered trademark of Xerox Corporation.
† DECnet and MicroVAX are trademarks of Digital Equipment Corporation.

First

computer corporation

645 blackhawk drive westmont, illinois 60559

Send me more information about First's compatibility connection.

Name

Position

Company

Address

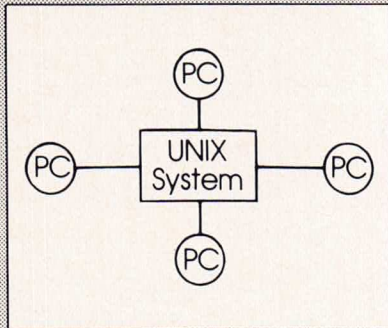
City State Zip

Phone

Call us First toll-free 1-800-292-9000 (In Illinois 1-312-920-1050)
TWX 910-651-1916 Sales Hours: 7:30 am — 6:30 pm CST
Sales Representatives in: Chicago, Dallas, Detroit, Houston, and St. Louis

PCworksTM WORKS!

Here at BASIS, we've checked PCworks carefully. And it does just what it claims to do: it makes your IBM PCTM (or compatible) a part of the UNIXTM revolution.



Now you and your PC can:

- Access UNIX programs
- Transfer PC & UNIX files
- Use the UNIX printer
- Back up PC files on UNIX
- Read, Create, & Send mail
- Access public & private networks

CALL or WRITE for
More Information

B.A.S.I.S.

UNIX Software, Hardware, & Timesharing

1700 Shattuck #1
Berkeley, CA 94709
(415) 841-1800

PCworks is a trademark of Touchstone Software
IBM PC is a trademark of IBM Corporation
UNIX is a trademark of Bell Laboratories

Circle No. 263 on Inquiry Card

Category	PCworks	Communique
Power	good	good
Flexibility	good	excellent
Ease of first time use	good	fair
Ease of long term use	excellent	excellent
UNIX Price	\$295	\$295
PC Price	\$195	\$195
Documentation	good	good

How PCworks and Communique stack up.

One minor annoyance I encountered with *Communique* was a screen-clearing bug that kept error messages from being properly erased, such that new messages sometimes overwrote old ones. This never caused me any confusion, but I'm something of a perfectionist about user interfaces and so hope that COSI's next released version fixes this.

Overall, it would seem that PCworks has a number of small advantages over *Communique*. So, is there any reason to consider *Communique*?

Yes! Although I would recommend PCworks for an end user who wants an immediate solution to immediate needs, there are several reasons why I would recommend consideration of *Communique* to others. One is that, taken together with *Visual Menu*,

the COSI program provides the potential for a considerable amount of customization. Although a review of *Visual Menu* itself will have to wait for a future column, I was sufficiently intrigued by its features and its elegant dovetailing with *Communique* that I would recommend *Communique* plus *Visual Menu* to anyone willing to do a certain amount of technical work to create a nicely integrated application package.

Another consideration is that COSI is a somewhat larger company whose additional technical staff implies (but does not guarantee) a greater ability to respond to customer bug reports and enhancement requests.

On the other hand, both companies are small enough to insure that there will be a significant time lag between minor bug reports and eventual fixes in new releases (major bug reports, though, usually preempt other activity in small companies). I do not see company size as a reason to avoid either product, however, since both companies provide good solutions to immediate needs.

Doug Merritt became one of the earliest UNIX users outside Bell Laboratories while attending UC Berkeley in 1976. He helped to debug termcap and contributed to the development of vi and curses. Mr. Merritt now works as a consultant in the San Francisco Bay Area. ■

UNIX*

Training in
DALLAS,
Austin and Houston
by

TELOS

1 Day Overview..... \$145

3 Day Fundamentals

with hands-on..... \$545

Call or Write for Schedule.

TELOS CONSULTING SERVICES

2915 LBJ, Suite 161
Dallas, Texas 75234
214/484-8672

*UNIX is a trademark of Bell Laboratories.

Circle No. 299 on Inquiry Card

PROBLEM SOLVER

Editing for programmers

by Bob Toxen

There are several features in **vi** that make editing program sources easier and faster. One of these features, called **autoindent**, automatically indents text for block structured languages such as C and Pascal. When **autoindent** is on, **vi** will remember how far the last line entered was indented and will automatically indent all subsequent lines by that same amount. If you need a larger indentation, simply enter the additional spaces or tabs as needed.

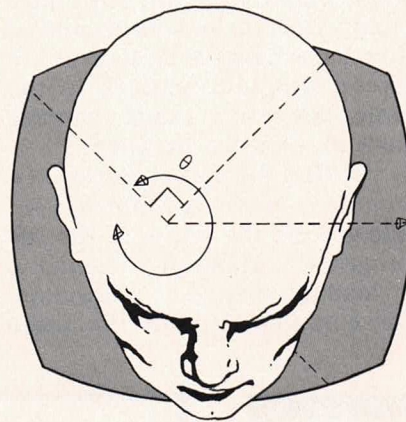
If you want to reduce the indentation for the current line (and subsequent lines), you can do so by entering one or more CTRL-Ds. Each CTRL-D causes **vi** to backup one full tab stop to the left. Preceding a CTRL-D with the digit zero (0) will cause **vi** to backup all the way to the left margin. You must adjust the indentation using CTRL-D *before* entering any text!

To turn on **autoindent**, issue the **vi** command:

```
:set autoindent
```

MAKING A MATCH

Another useful **vi** feature programmers can use is **showmatch**, for help in keeping track of matching parentheses and curly braces. When **showmatch** is on, **vi** will search backwards through text when you enter a right parenthesis (}) or right curly brace (}) to find the matching left parenthesis



or left curly brace. If one doesn't exist, **vi** will ring the terminal bell to alert you to a probable error.

If **vi** finds the matching parenthesis or brace on the current screen, it will move the cursor to this matching parenthesis or brace briefly (by doing a **sleep(1)**) to reassure you that you have not made an omission.

Also, when you enter a search pattern that includes a right parenthesis or curly brace but excludes the matching item, **vi** will ring the bell. The **showmatch** feature is turned on by issuing the command:

```
:set showmatch
```

If you position the cursor on a left or right parenthesis, curly brace or square bracket ([]) or ([]) and give the percent command (%), **vi** will move the cursor to its mate (or ring the bell if it can't

make a match). The cursor is left at the matching item and this becomes the new cursor position.

When **vi** searches for matching parentheses, braces or brackets, it does not care whether or not it finds the mate in code, comments or literal strings. The **vi** editor is not that smart – yet.

Compilers will frequently give the line number that an error occurs on. By using **vi**'s **G** command, you can quickly move to that specified line number. Simply precede the "G" with the number of the line you want the cursor to move to. Thus to move to line 137, issue the command:

```
137G
```

You can also use the CTRL-G command to determine what line you are currently on. When you do, editing changes will be written to the buffer file on disk so that if the system subsequently crashes the latest version of the file—with all editing changes intact—can be recovered later. (The **vi** command periodically writes editing changes automatically to the buffer to prevent the loss of more than a few lines of changes.)

On large programming projects, source is generally divided into many files. As a result, it can sometimes be a chore just to find the file that contains a particular function you want to work with. This was once a problem only **grep** could grapple with.

The program **ctags** and **vi**'s

PROBLEM SOLVER

tags command have since been created, though, to make life easier. To use them, invoke **ctags** with the name of each source file on the command line. For example:

```
ctags *.c
or:
ctags *.p
```

The **ctags** command will parse the source files looking for functions and storing the name of each one and the file where it occurs in a datafile named **tags**. By using a search pattern instead of a line number, the function can still be found if lines are added to or deleted from the file. The **ctags** command knows how to handle C, Pascal and FORTRAN programs. It only needs to be invoked at the outset and subsequently whenever functions are added.

If you want to edit the function **doecho** then invoke **vi** with the **-ta** flag and **doecho** as its

parameter, as in the following example:

```
vi -ta doecho
```

The **vi** editor will search through the **tags** file to see what file **doecho** occurs in, edit that file, search through the file for the start of the **doecho** function and move the cursor to that point. You can then begin editing.

If you then want to edit the function **dinit**, you can give the **vi** command:

```
:ta dinit
```

Since every C program must have a main function and one should be able to use **ctags** even if there are several program sources in the same directory, **ctags** handles main in a special manner.

To find the main function of a program, enter the letter "M" followed by the name of the file (program) without any trailing **.c** or leading directory pathname. For example, to edit the **main**

function of the **date** program (which is in **date.c**), give the command:

```
vi -ta Mdate
```

The **tags** feature can be used for other purposes than editing C, Pascal and FORTRAN sources. The **tags** file is just a database that can be created by any means, wherein each entry takes up one line and consists of three fields separated by tags. The first field is the name of the item as you will specify it to the **tags** command. The second field is the file it occurs in. The final field is the search pattern **vi** should use to find the item. To do this, any **vi** command may be used.

*Bob Toxen is a member of the technical staff at Silicon Graphics, Inc. He has gained a reputation as a leading **uucp** expert and is responsible for ports of System V for the Zilog 8000 and System III for the Motorola 68000.*

THE HARDWARE/SOFTWARE SOLUTION FOR UNIX V

Convergent Technology Mini-Frame OEM (32 bit — 8 User System)

- Up to 2M Bytes Memory
 - 26MB, 50MB or 140MB Disk
 - RS-232, RS-422 ports
 - PT-100 Intelligent Terminal
 - GT-100 Intelligent Graphics Terminal
 - Program Languages: CTIX™C, Cobol, Basic, Fortran, Pascal, RM Cobol, RM Fortran
- 30 Day Availability

Unix V Software

- Virtual Memory/Berkley Enhancements
- Accounting (includes AR AP GL & Inv.)
- Spread Sheet
- Word Processing
- CAD (Computer Aided Design)
- Relational Data Base
- IBM PC Networking

We will convert your Cobol, Dibol or Fortran & Basic programs to run on Convergent Mini-Frames.

Call today for prices and literature
Excellent lease programs available

AMCOM

SOFTWARE INC.

5200 W. 73RD ST. • MINNEAPOLIS, MN 55435
(612) 835-4737

troff | tplus > /dev/*

TPLUS™ processes the output from TROFF† for a wide variety of devices. In *preview mode*, you see exactly what the final typeset output will look like, using a high-resolution display or a daisy-wheel, dot matrix, or laser printer. In *native mode*, TPLUS produces output of optimal quality from these same devices, useful for draft, review, or low-volume production.

Of course, TPLUS drives all modern phototypesetters. TPLUS drivers are fully functional, offering many useful features for production typography.

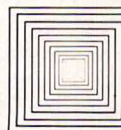
TPLUS was designed for device independence from the start. Whenever a suitable output device is available, TPLUS can be adapted to it. The manufacturers whose equipment is driven by TPLUS are:

- *phototypesetters*: CG, Mergenthaler, Autologic
- *laser printers*: Xerox, Hewlett-Packard
- *letter-quality printers*: Diablo, Qume, NEC
- *dot matrix printers*: Epson, C. Itoh
- *high-resolution displays*: Teletype, Sun, BBN

For further information on TPLUS and our other TROFF-related offerings (including the UNI•TEXT typesetting service), please write or call.

TPLUS is a trademark of Textware Intl.

† All versions—incl. v7, Sys III, Sys V, & DWB.



**TEXTWARE
INTERNATIONAL**

P.O. BOX 14 • HARVARD SQUARE STATION
CAMBRIDGE, MASSACHUSETTS 02238 USA

TELEPHONE
(617) UNI•TEXT

THE UNIX GLOSSARY

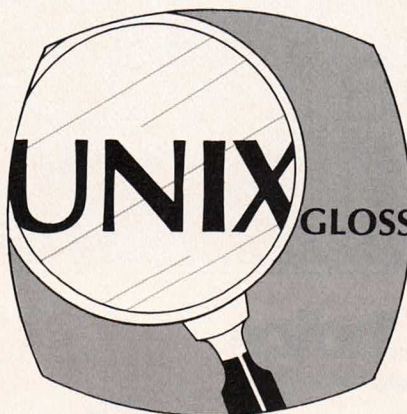
A guide to Writer's Workbench

by Steve Rosenthal

abst – a program that reports the percentage of abstract words in a document (defined, for this purpose, as words contained in a list of several hundred terms commonly used to describe non-concrete concepts). Based on the text samples used during the development of Writer's Workbench, documents that register a percentage of 2.3 or higher are considered to be a possible problem.

acro – a program that finds "acronyms" in the input file, listing out the lines in which they occur so you can make sure you have defined them. Actually, the program finds all words with two or more adjacent capital letters, not all of which are acronyms. Note also that not all acronyms are necessarily capitalized. However, the program is still useful for spotting instances where alphabet-soup words might have been used without adequate explanation.

ddict – the name of a file containing phrases you want added to those marked by the Writer's Workbench **diction** program as likely candidates for rewriting. It can also contain standard phrases you want left unmarked. You create **ddict** with **dictadd**, normally locating it in **\$HOME/lib/wwb** (where **\$HOME** is your login directory).



dictadd – a utility you run to add words or phrases to various lists the package uses to proofread your documents. This utility is most often invoked to edit **ddict** (your personal phrase list used by the **diction** program), **sexdict** (your personal list for the **sexist** program), **spell**dict (your personal spelling list) and files that all these programs can use instead of the standard system phrase or word lists.

diction – a program that brackets phrases in your document that match entries in a list of oversed, frequently misused or problem phrases. It can also supply suggested replacements from its list, but as the program has no means of reading context, some of these suggestions are less valuable than others. The **diction** program can

be used separately, and is invoked by the **proofr** program (which, in turn, is invoked by **wwb**).

double – a program that checks your document for two or more adjacent repetitions of the same word (usually an error). The program can be used by itself and is invoked by **proofr** (which, in turn, is invoked by **wwb**).

findbe – a program that outputs a copy of a document with all instances of the verb "to be" capitalized and underlined. Because these verbs are often used excessively and are frequently part of misused constructions such as passive sentences, reading over a copy of the document with these verbs highlighted can often call attention to problems during editing.

match – a program that outputs statistics about document form and structure in a format that allows easy comparison between documents. It also allows for quick scanning of the Writer's Workbench values for a single document. This program takes the tables produced by **style** for input but not for the document files themselves – which may at first seem confusing. If you're an expert with the **awk** pattern-matching program, you can easily

Scribe[®] supports laser printers

You've heard a lot about laser printers . . . high print quality, attractive fonts, fast, quiet, affordable. But you haven't heard a lot about software to support them.

The **Scribe Document Production System** supports more laser printers than any other software. Laser printers like the DEC LN01™, the Xerox 2700, 8700, 9700, the Symbolics™ LGP-1, and all Imagen and QMS Lasergrafix™ laser printers.

Buying a laser printer? No matter which one you choose, you'll get the most from it with **Scribe**.

For more information, contact
UNIOLOGIC, Ltd.
160 North Craig Street
Pittsburgh, PA 15213
412-621-2277

Scribe document production software is available for DEC 10, 20, and VAX, Prime, IBM mainframes, the Apollo and Sun workstations. "Scribe" is a registered trademark of UNIOLOGIC, Ltd.



Circle No. 272 on Inquiry Card

The Leader's Choice



Information Industries, Inc. (Triple-I) has been the leader's choice in technical Skills Management since 1971.

By employing a highly skilled team of professionals, known as Systems Engineers, Triple-I insures many of the world's top corporations are equipped to meet the information/data processing demands of the future.

These professionals, with expertise in areas like UNIX* and other software development tools, have welcomed the technically challenging growth opportunities we have to offer. They enjoy the:

- increased challenge
 - enhanced recognition
 - and accelerated growth
- that is associated with being a member of Triple-I's team.

Along with the emergence of the UNIX based systems, comes Triple-I's demand for technically competent professionals to design, develop and maintain these systems.

Call 1-800-TRIPLE I—ext. 4100, to learn why Triple-I has become The Leader's Choice.

* UNIX is a registered trademark of Bell Labs

Information Industries Inc. • 8880 Ward Parkway • Kansas City, MO 64114
Circle No. 273 on Inquiry Card

GLOSSARY

write your own customized equivalent.

mkstand — a program that compiles a set of standards for evaluating text from a set of sample documents you provide. With **mkstand**, you can create your own alternate standards for use by the Writer's Workbench **prose** program (for checking document readability, sentence structure and style) instead of using the default standards developed by Bell Laboratories.

org — a program that outputs only titles and the first and last sentence from each paragraph of input text. By looking at the document in this abbreviated form, you can more easily see the overall structure and organization, and check the transitions between paragraphs.

parts — the utility used by the Writer's Workbench **style** program to classify words in text by grammatical parts of speech. This utility is automatically invoked by **style**, but you can run it by itself if you want to see what sort of input it feeds to **style**.

procedural document — a text-file containing step-by-step instructions on how to accomplish a task. This format is common in instruction manuals, but the programs in Writer's Workbench that check writing style do not work well when applied to this type of text — as it is normally quite different in linguistic structure and form than the expository text the style-checking programs expect. However, various proofreading programs in Writer's Workbench (such as **punct** and **spellwwb**) will do fine with this sort of document.

proof — an alternate name (or **alias**) for **proofr**, the Writer's Workbench program that proof reads text. See **proofr** for more details.

proofr — one of the two main subprograms in Writer's Work-

Circle No. 115 on Inquiry Card ➤

E PLURIBUS UNIX*

Out of many—there is UNIX

E PLURIBUS UNIX EXPO

Out of many—there is . . .

UNIXTM EXPO **THE UNIX OPERATING SYSTEM
EXPOSITION & CONFERENCE**

October 16, 17, 18, 1984

Sheraton Centre Hotel, New York City—Conference

Marina Expo Complex, New York City—Exposition

The comprehensive, practical, business/learning event designed solely and specifically to address the myriad technical and business aspects of the UNIX OPERATING SYSTEM. This national trade show is created to bring together manufacturer, ISO and sophisticated end-user in order to facilitate the transaction of business in this dynamic field. Exhibiting in UNIX EXPO will ensure that your firm can position itself in the vanguard of the UNIX revolution, and claim your share of the market dollar.

The UNIX Exposition Experience

State-of-the-art exposition featuring more than 350 exhibit booths of UNIX related hardware, software, peripherals, professional associations, trade publications, and service organizations.

An innovative, on-site Job Fair hosted by PENCOM SYSTEMS, INC., the nationally recognized leader in UNIX recruiting, will permit exhibiting firms to disseminate information concerning employment opportunities. Career objectives can be discussed, and meetings with company representatives scheduled.

Penetrating, multi-track conference program focusing on major technical, marketing, and sales aspects of UNIX. Coordinated by noted UNIX advocate James Joyce, the program will assemble industry leaders to address topics of vital interest to all segments of the UNIX environment.

New York City—the core of the largest computer marketplace in the world. National Expositions, a leader in show management for more than three decades will conduct a massive audience promo program to ensure maximum attendance of qualified UNIX resellers and sophisticated end-users.

**THE COMPLETE UNIX EXPO EXPERIENCE IS A MUST FOR:
MANUFACTURERS WISHING TO SELL,
ISO'S, TECHNICAL PERSONNEL AND WHOLESALERS WISHING TO LEARN,
END-USERS WISHING TO EXAMINE, COMPARE, AND SELECT.**

For Complete details regarding exhibit space availability
or attending UNIX EXPO contact:

UNIXTM EXPO

National Expositions Co., Inc.

14 West 40th Street, New York, NY 10018.

Telephone 212/391-9111 Telex: 135401 DIMCOMM

bench, **proofr** checks for misspelled or doubled words, erroneous punctuation, misused phrases and split infinitives. It can be run by itself and is automat-

ically invoked by **wwb**. Most of the work in **proofr** is done by invoking **spellwwb**, **double**, **punct** and **diction**.

prose - a program that gives

you a narrative description of the result of a check of your document's style. The **prose** program first invokes **style** to collect statistics about such factors as readability level, sentence length and sentence type. Then **prose** presents and explains the statistics.

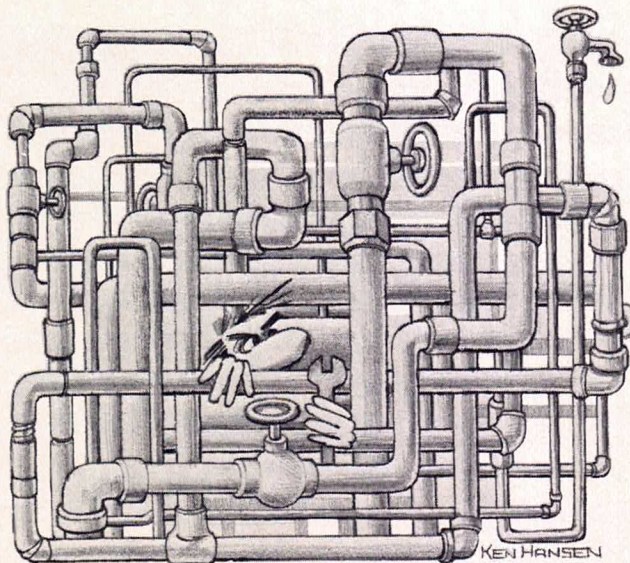
punct - a program that checks a text document for errors in punctuation. If you run the program directly, it will create as output a copy of your file with any detected punctuation errors corrected. The program is also invoked by **proofr** (which in turn is invoked by **wwb**), but when run automatically it does not create the corrected output file but only a listing of suspected errors.

punctrules - a program that displays the rules used by the Writer's Workbench **punct** program to check punctuation in textfiles. It is mainly used to find out why the **punct** program has objected to a certain construction.

sexdict - a personal list of gender-specific terms you can have the Writer's Workbench **sexist** program check for in textfiles while it looks for other words included in the standard **sexwords.d** list. You create **sexdict** (normally located in **lib/wwb** under your login directory) with the **dictadd** program. As with items on the standard list, your own terms can be words or phrases, and can include suggested replacements.

sexist - a program that scans input text files for gender-specific words and phrases. By providing a listing of lines that include these terms along with a listing of suggested replacements, it makes avoiding sexist terms easier.

spelladd - a program that you can use to add words to your personal spelling dictionary so **spellwwb** will not flag those terms as probable misspellings.



Without SoftShell,[™] learning to live with UNIX[™] could be a real trap.

SoftShell is a convenient interface which guides you gracefully through UNIX. If you're new to UNIX, or have the task of training new users, SoftShell simplifies its complexity. If you're already a UNIX fan, you'll love SoftShell because it augments your system. You'll find yourself further exploring the great depth and versatility of UNIX. Available for UNIX System-V, Berkeley 4.2 and IBM's PC/IX, SoftShell is an invaluable tool for users at all levels of expertise. For information, contact Logical Software, Inc., 17 Mount Auburn Street, Cambridge, MA 02138, (617) 864-0137.

UNIX is a trademark of AT&T Bell Laboratories.



Logical Software Inc.

Circle No. 274 on Inquiry Card

The **spelladd** program creates or adds to your file **spelldict** (located in **lib/wwb** in your login directory) words listed on the command line. You can also add to your spelling list interactively with the **dictadd** command.

spelldict – a personal list of words you want the Writer's Workbench spell-checking program to accept as it scans documents for unrecognized (and therefore possibly misspelled) words. This file is normally located in **lib/wwb** under your login directory. You add words to the list with **spelladd** or **dictadd**.
spelltell – a standalone program that outputs words that match a particular character pattern. By supplying a pattern consisting of part of a word that you're not sure how to spell, you can get a list of words in the Writer's Workbench dictionary that match the pattern. The pattern matching uses the **grep** utility, so you can specify letters at the start, end or any other position in the word. However, the system does not use Soundex or other sound-matching system, so if you miss even one letter in your pattern specification, you won't find your word.

spellwwb – a program that checks the words in a document against a standard system dictionary and an optional user dictionary, listing as possible errors any words that are not found and not derivable by adding common prefixes and suffixes to the listed words. Because the program uses rules and a dictionary of moderate size, it may accept some plausible constructions that are not words and yet fail to find others that are spelled legitimately. The **spellwwb** program is automatically invoked by **proofr** (which, in turn, is invoked by **wwb**), but you can also run it by itself.

splitrules – a program that displays the rules concerning split

infinitives. The **proofr** program marks split infinitives as it checks textfiles, and **splitrules** explains the grammatical basis.

style – one of the principal text-checking programs in the Writer's Workbench package. The **style** program creates a table displaying statistics about the structure and format of text, including readability indexes, sentence lengths, sentence types, abstractness, verb types and types of words used to start sentences. The **style** program will also, on command, output sentences that meet certain test criteria. It can be used alone for gathering statistics about a document. When invoked by **prose** (which, in turn, is invoked by **wwb**), **style** collects the statistics that **prose** presents and explains them.

topic – a program that outputs a list of the 20 most-used nouns and adjective-noun pairs in an input document. Its purpose is to help identify, index and abstract articles. Unfortunately, **topic** works strictly on words as character strings, ignoring all meaning and content. Also note that the default settings remove **mm** macros and list items before inspecting the text, so terms in these portions of the text will not be counted.

wwb – the name of the main Writer's Workbench program. It

invokes **proofr** to check spelling, punctuation and diction, plus **prose** to check document style and readability.

wwbaid – a program that provides online tutorial and reference information. Various options offer a general introduction and tutorial, a list of all the Writer's Workbench commands and their functions and syntax, a list of published papers about the package, and a summary of the manual sections for each command.

wwbhelp – a program that presents information on commands related to topics selected from a list of commands and concepts. Like most Writer's Workbench query programs, you must know the exact topic and its spelling (or search through the list provided by the program) to get the right listing.

wwbinfo – a program that outputs a table summarizing the available commands and dictionaries along with a short explanation of their functions. This same information is presented in several other forms by **wwbaid** and **wwbhelp**.

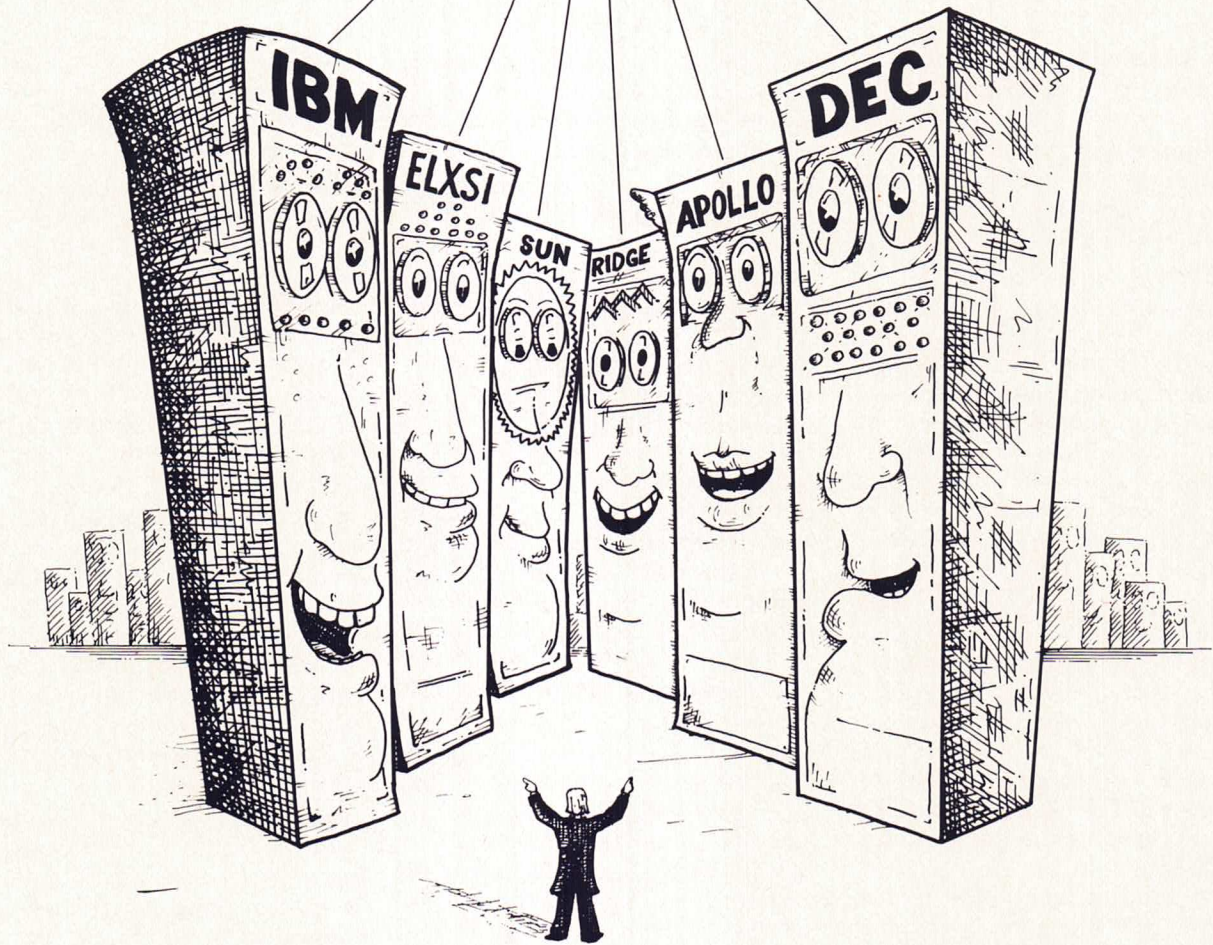
Steve Rosenthal is a lexicographer and writer living in Berkeley. His columns regularly appear in six microcomputer magazines. ■

SPECIAL PRICE FOR PLEXUS COMPUTERS

"2" • P-60 - P35 with 2MB; 80MB Disc;
2nd ICP Software includes:
SYS III Utility, FORTRAN, BASIC, UNIFY
• With factory warranty.

For information
Call: (213) 643-6997 Marketing

MAINSAIL™



SPEAK MAINSAIL. BECAUSE NOT EVERYONE SPEAKS UNIX.

Most computer programmers like the UNIX environment. It gives them a convenient set of software development tools. There's one problem, however. Applications must often run on machines that don't have UNIX.

That's where MAINSAIL comes in.

MAINSAIL is a powerful programming language that can help cut your development time and eliminate software conversion costs. You'll be able to take advantage of the development power of UNIX, while retaining the ability to move to other systems. And you'll be amazed at how easy it is to learn this proven, versatile language.

If you like UNIX, but need to keep your portability options open, just let us know. We'll show you how MAINSAIL can accelerate your development of portable, sophisticated application programs—not only under UNIX, but under VAX/VMS®, VM/CMS®, and a variety of other operating systems. For details, contact us at: XIDAK, Inc., 530 Oak Grove Ave., Suite 101, Menlo Park, CA 94025, (415) 324-8745.

UNIX is a trademark of Bell Laboratories.
VAX/VMS is a trademark of Digital Equip. Corp.
VM/CMS is a trademark of IBM Corp.

xidak

CALENDAR

AUGUST

July 30 - August 3 Uni-Ops Conference for UNIX and C Users, San Francisco, CA, Including (August 2-3) UNIX Software & Services Exposition. Contact: Uni-Ops, PO Box 27097, Concord, CA 94527-0097. 415/689-4382.

August 28 Uni-Ops Monthly Meeting, Palo Alto, CA: Expert panel discussion sponsored by the non-profit UNIX user group. "UNIX and IBM PC s." Contact: John Bass, 408/996-0557 or Paul Fronberg, 408/988-1755.

SEPTEMBER

September 11-14 UNIX Systems Expo/84, Los Angeles, CA. Contact: Computer Faire, Inc., 611 Veterans Boulevard, Redwood City, CA 94063. 415/364-4294, or CFI in Newton, MA, 617/965-8350.

OCTOBER

October 16-18 UNIXEXPO, The UNIX Operating System Exposition, New York, NY. Contact: National Expositions Co., Inc., 14 West 40th Street, New York, NY 10018. 212/391-9111.

NOVEMBER

November 14-18 Comdex, Las Vegas, NV. Contact: The Interface Group, Inc., 300 First Avenue, Needham, MA 02194. 617/449-6600.

TRAINING CALENDAR

AUGUST

August 2-3 Computer Technology Group, Boston, MA & Washington, DC: "Using Advanced UNIX Commands." Contact: CTG, Telemedia, Inc., 310 S. Michigan Ave., Chicago, IL, 60604. 800/323-UNIX, or in Illinois, 312/987-4082.

August 6 AT&T Technologies, Hopewell, NJ: "UNIX System Device Drivers." Contact: AT&T Technologies, Corporate Education & Training, PO Box 2000, Hopewell, NJ 08525. 800/221-1647.

August 6 AT&T Technologies, Hopewell, NJ: "UNIX System/C Language Interface." Contact: AT&T (see previous entry).

August 6-10 Bunker Ramo Information Systems, Trumbull, CT: "Intro to UNIX & C." Contact: Bunker Ramo Information Systems, Training Services Group, Trumbull Industrial Park, Trumbull, CT 06609.

August 6-10 Computer Technology Group, Boston, MA & Washington, DC: "UNIX Internals." Contact: CTG (see August 2-3).

August 7-9 Computer Technology Group, Dallas, TX: "UNIX Administration." Contact: CTG (see August 2-3).

August 8-10 Digital Seminar Program, Seattle, WA: "UNIX Operating System Overview." Contact: Digital Educational Services, 12 Crosby Dr., Bedford, MA 01730. 617/276-4949.

August 13 AT&T Technologies, Hopewell, NJ: "Fundamentals of the UNIX Operating System for Users." Contact: AT&T (see August 6).

August 13 AT&T Technologies, Lisle, IL: "UNIX System Internals." Contact: AT&T (see August 6).

August 13 AT&T Technologies, Dublin, OH: "UNIX System Administration." Contact: AT&T (see August 6).

August 13-17 Bunker Ramo Information Systems, Trumbull, CT: "Advanced C." Contact: Bunker Ramo (see August 6).

August 13-14 Computer Technology Group, Dallas, TX: "Advanced C Programming Workshop." Contact: CTG (see August 2-3).

August 14-16 Computer Technology Group, Boston, MA & Washington, DC: "UNIX Administration." Contact: CTG (see August 2-3).

August 15-17 Digital Seminar Program, Seattle, WA: "The C Programming Language." Contact: Digital Educational Services (see August 8-10).

August 15-17 Computer Technology Group, Dallas, TX: "Advanced C Programming Under UNIX." Contact: CTG (see August 2-3).

August 16 AT&T Technologies, Hopewell, NJ: "Shell Command Language for Users." Contact: AT&T (see August 6).

August 20 AT&T Technologies, Lisle, IL: "C Language for Experienced Programmers." Contact: AT&T (see August 6).

August 20 AT&T Technologies, Sunnyvale, CA: "UNIX System Internals." Contact: AT&T (see August 6).

August 20-21 Computer Technology Group, Boston, MA & Washington, DC: "Advanced C Programming Workshop." Contact: CTG (see August 2-3).

August 20-24 Bunker Ramo Information Systems, Trumbull, CT: "Advanced UNIX." Contact: Bunker Ramo (see August 6).

August 20-24 Computer Technology Group, Dallas, TX: "Berkeley UNIX Fundamentals and 'csh' Shell." Contact: CTG (see August 2-3).

August 20-24 Plum Hall Training, Raleigh, NC: "Advanced C Topics." Contact: Plum Hall, 1 Spruce Ave., Cardiff, NJ 08232. 609/927-3770.

August 22 AT&T Technologies, Hopewell, NJ: "Fundamentals of the UNIX Operating System for Programmers." Contact: AT&T (see August 6).

August 22-24 Computer Technology Group, Boston, MA & Washington, DC: "Advanced C Programming Under UNIX." Contact: CTG (see August 2-3).

August 27 AT&T Technologies, Hopewell, NJ: "Shell Command Language for Programmers." Contact: AT&T (see August 6).

August 27 AT&T Technologies, Lisle, IL: "UNIX System/C Language Interface." Contact: AT&T (see August 6).

August 27-31 Computer Technology Group, Boston, MA & Washington, DC: "Berkeley UNIX Fundamentals and 'csh' Shell." Contact: CTG (see August 2-3).

August 28 AT&T Technologies, Sunnyvale, CA: "UNIX System Device Drivers." Contact: AT&T (see August 6).

August 28 Computer Technology Group, Dallas, TX: "UNIX Overview." Contact: CTG (see August 2-3).

August 29-31 Computer Technology Group, Dallas, TX: "UNIX Fundamentals for Non-Programmers." Contact: CTG (see August 2-3).

SEPTEMBER

September 5 AT&T Technologies, Hopewell, NJ: "UNIX System Screen Editor vi." Contact: AT&T (see August 6).

September 5 AT&T Technologies, Sunnyvale, CA: "Fundamentals of the UNIX Operating System for Programmers." Contact: AT&T (see August 6).

September 6 AT&T Technologies, Lisle, IL: "Overview of the UNIX System." Contact: AT&T (see August 6).

September 10 AT&T Technologies, Sunnyvale, CA: "Shell Command Language for Programmers." Contact: AT&T (see August 6).

September 10 AT&T Technologies, Lisle, IL: "Fundamentals of the UNIX Operating System for Users." Contact: AT&T (see August 6).

September 10 AT&T Technologies, Hopewell, NJ: "UNIX System Document Preparation." Contact: AT&T (see August 6).

September 10-12 Computer Technology Group, San Francisco, CA: "UNIX Fundamentals for Non-Programmers." Contact: CTG (see August 2-3).

September 10-14 Bunker Ramo Information Systems, Trumbull, CT: "Advanced C." Contact: Bunker Ramo (see August 6).

September 11 Computer Technology Group, New York, NY & Washington, DC: "UNIX Overview." Contact: CTG (see August 2-3).

September 12-14 Computer Technology Group, New York, NY & Washington, DC: "UNIX Fundamentals for Non-Programmers." Contact: CTG (see August 2-3).

September 13 AT&T Technologies, Lisle, IL: "Shell Command Language for Users." Contact: AT&T (see August 6).

September 13-14 Computer Technology Group, San Francisco, CA: "Shell as a Command Language." Contact: CTG (see August 2-3).

September 17-19 Computer Technology Group, New York, NY & Washington, DC: "UNIX Fundamentals for Programmers." Contact: CTG (see August 2-3).

September 17-21 Bunker Ramo Information Systems, Trumbull, CT: "Intro to UNIX." Contact: Bunker Ramo (see August 6).

September 17-21 Computer Technology Group, San Francisco, CA: "C Language Programming." Contact: CTG (see August 2-3).

September 20-21 Computer Technology Group, New York, NY & Washington, DC: "Shell as a Command Language." Contact: CTG (see August 2-3).

September 24 AT&T Technologies, Sunnyvale, CA: "C Language for Experienced Programmers." Contact: AT&T (see August 6).

September 24 AT&T Technologies, Lisle, IL: "Software Development Under the UNIX System." Contact: AT&T (see August 6).

September 24 AT&T Technologies, Lisle, IL: "UNIX System Device Drivers." Contact: AT&T (see August 6).

September 24-25 Computer Technology Group, San Francisco, CA: "Shell Programming." Contact: CTG (see August 2-3).

September 24-28 Bunker Ramo Information Systems, Trumbull, CT: "C Programming." Contact: Bunker Ramo (see August 6).

September 24-28 Computer Technology Group, New York, NY & Washington, DC: "C Language Programming." Contact: CTG (see August 2-3).

September 25 Computer Technology Group, Chicago, IL: "UNIX Overview." Contact: CTG (see August 2-3).

September 26 AT&T Technologies, Lisle, IL: "UNIX System Tools." Contact: AT&T (see August 6).

September 26-28 Computer Technology Group, Chicago, IL: "UNIX Fundamentals for Non-Programmers." Contact: CTG (see August 2-3).

September 26-28 Computer Technology Group, San Francisco, CA: "Using Advanced UNIX Commands." Contact: CTG (see August 2-3).

Dear Readers: If you or your company are planning events of interest to the UNIX community, please send announcements to:

*UNIX Review Calendar
International Technical Seminars
520 Waller Street
San Francisco, CA 94117*

Please include sponsor, date and location of event, address of contact and relevant background information.

SOFTWARE PROFESSIONALS COME GROW WITH US

MicroLabs™ has several opportunities for experienced professionals at its Houston, Texas facilities. We are involved in software development for the office automation areas, and have a highly sophisticated team of professionals working in the field of advanced technology using UNIX and "C" language.

Current openings:

- * **SOFTWARE ENGINEER**
with UNIX; "C"; VAX 750/780 experience
- * **GRAPHICS SPECIALIST**
with DI-3000, PTI, Versaplot, Calcomp or similar graphics experience; "C"; business graphics applications; UNIX
- * **DATABASE SPECIALIST**
with Database internals; "C"; IDMS, dBasell or related database; UNIX or CICS experience
- * **QUALITY ASSURANCE TEST ENGINEER**
with software testing; analysis of product specifications and identification of test criteria; UNIX; "C"

If excellent salaries and benefits, including stock options, a friendly atmosphere, informal dress, private offices, and an opportunity to grow with a dynamic company appeal to you, send us your resume now to:

Circle No. 275 on Inquiry Card

MICROLABS

6300 Richmond, Third Floor
Houston, Texas 77057
Attn: Business Office Manager

UNIX™ is the trademark of Bell Laboratories, Inc.

LOTUS 1-2-3* MOVE OVER QUALITY SOFTWARE ANNOUNCES Q-CALC (version 3.0)

- spreadsheet
- data management
- forms processing
- graphics

PLUS
it runs on UNIX

For more information write/call

Quality Software Products
348 S. Clark Dr.
Beverly Hills, CA 90211
213-659-1560

*Lotus 1-2-3 is a trademark of Lotus Development Corp.

Circle No. 52 on Inquiry Card

ADVERTISERS INDEX

Accucom Data	36	Logical Software	102
Aim Technology	37	Manx Software	25
Amcom Corp.	98	Mark Williams	17
Applix	9	Medical Informatics	93
AT&T	52	Microlabs	106
BASIS	96	Microware Systems	7
BDS	92	MIPS Software	68
Cadmus	108, Cover III	NCR	85
Codata	67	Network Consulting	87
Computer Cognition	93	Oasys	49
Computer Faire	57	Quality Software	106
Computer Methods	11	Relational Database Systems	1, 2, 3
Computer Technology Group	33	Relational Technology Inc.	12
Convergent Technology	Center Spread	Reliable Data	65
COSI	43	Rhodnius	91
CSSI Corp.	78	Santa Cruz Operation	71
Emerald City Inc.	63	Scientific Placement	65
Faircom	84	Silogic	53
First Computer Corp.	95	Software Results	29
Gould	19	Teletype	27
Group III Electronics	103	Telecon	68
Handle Corp.	Cover II	Telos Consulting Services	96
Heurikon	50	Textware	98
Hewlett-Packard	22, 23	Unilogic	100
Human Computing Resources	13, 15	Unipress Software	79, 81
Human Designed Systems	51	Uniq Digital	45
I.B.C.	Cover IV	Unisoft	41
Image Network	32, 34	UNIX EXPO	101
Information Industries Inc.	100	User Training Corp.	89
ITS	90	WYSE Technology	35
Lantech	76	Xidak	104
LMC	83		

Coming up in September

- Preventative System Medicine
- Error Modeling
- Turnkey System Administration
- Security Measures
- Horror Stories
- Making uucp Work
- Man in the Street Interview





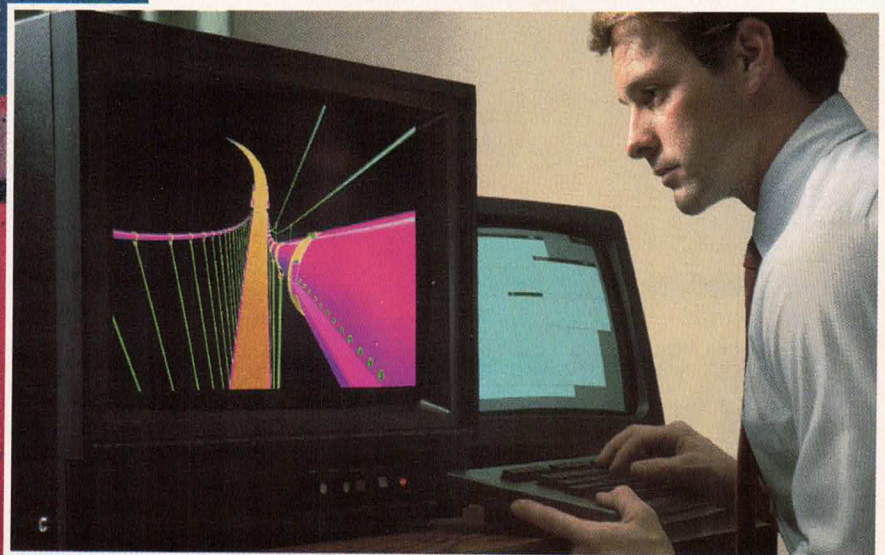
With a complete set of tools, you can build almost anything.

While other companies offer OEMs a few tools to build with, Cadmus provides the complete set.

For example, the Cadmus 9000 combines all the power of a mainframe, the speed of a mini, the price of a micro and the unlimited application flexibility of UNIX™ System V with Berkeley 4.2 enhancements.

All in a distributed environment.

In fact, the system's unique UNISON™ networking software provides the full resources



of the entire network to any computational node through network-wide virtual memory with demand paging.

A 10 Mbit Ethernet™ or 50 Mbit fiber optic LAN serves as the interprocessor bus.

And comprehensive graphics, development and decision support tools facilitate graphics applications in a windowed, multiple process environment.

In addition, we support every Cadmus 9000 with Thunderbolt™ service that puts an engineer at your door anywhere in the country within 24 hours. Guaranteed.

To get the complete picture on this high powered graphics-oriented distributed system, call Cadmus at 617-453-2899.

The Cadmus 9000. Finally, you've got a complete set of tools.

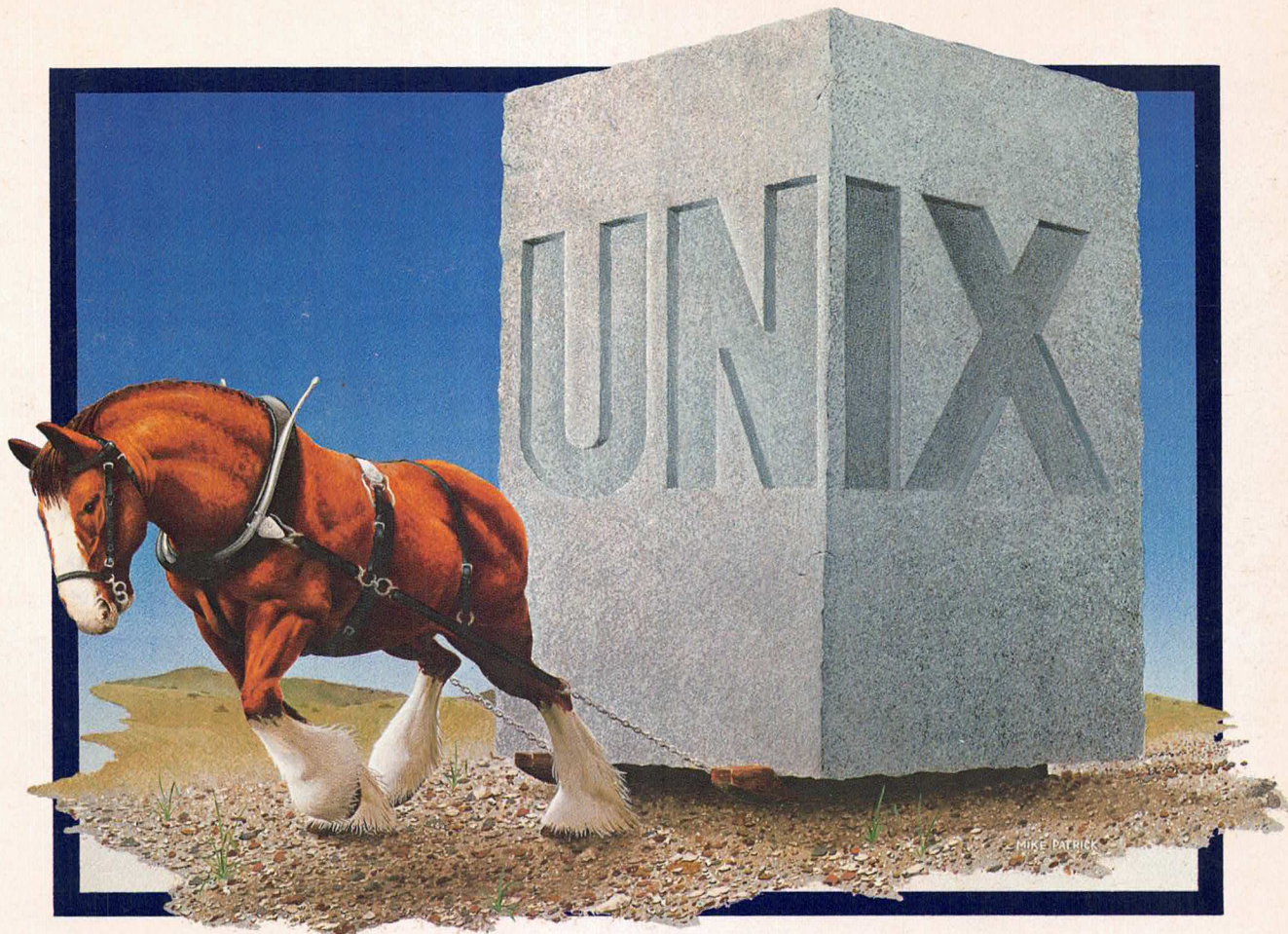
CADMUS
COMPUTER SYSTEMS

The Distributed
Mainframe.™

600 Suffolk Street, Lowell, MA 01854

USENET: ...!wivax!cadmus TELEX: 948257

UNIX is a trademark of Bell Laboratories. Ethernet is a trademark of Xerox Corporation.



UNIX™ HORSEPOWER!

There are a lot of UNIX based systems on the market today claiming to be "SUPERMICROS". But do they really have what it takes to run multi-user UNIX well? The IBC ENSIGN™ does and here's why:

FAST MEMORY: No computer running at **any** clock speed can run faster than it's overall memory design. The ENSIGN has up to 8MB of 120nsec memory with dual bit error correction. With IBC's proprietary memory management, **all** of this memory runs with no wait states as fast as the 68000 CPU will go. Compare this to other systems running only small cache memories at full speed. Other multiple user systems cannot load all their programs into a small cache memory. Their systems slow down considerably under a heavy multi-user load.

INTELLIGENT SERIAL I/O CONTROLLER: Even the fastest CPU will slow down when it's trying to handle interruptions from multiple on-line users. The ENSIGN provides slave serial I/O CPU's **and** FIFO buffering for both input and output. The result is the ENSIGN's ability to support up to 32 users, with heavy serial I/O demand, while leaving the main 68000 CPU free to run with little serial I/O overhead.

INTELLIGENT DISK CONTROLLER AND HIGH PERFORMANCE DISK DRIVES: The ENSIGN has a slave CPU to handle all disk operations, **plus** 16K of disk buffering. IBC's proprietary disk DMA allows high speed data transfer to main memory without slowing down the main CPU. Further, the ENSIGN

supports SMD type 8" hard disks with much faster seek times and transfer rates than 5¼" hard disks usually found in personal desk top computers.

THE RESULTS: The IBC ENSIGN runs multi-user UNIX at performance levels not attainable by other supermicros.

Call IBC and get a copy of IBC's multi-user benchmarks—benchmarks that test 8 users running large CPU programs, with heavy disk I/O **and** heavy serial I/O simultaneously. You'll find that nothing can compare to the ENSIGN.



If you want to run multi-user UNIX on a high performance system with up to 32 users, 8MB memory, and over 1,000MB disk storage, see the IBC ENSIGN.

Outside The USA

IBC/Integrated Business Computers

21621 Nordhoff Street
Chatsworth, CA 91311
(818) 882-9007
Telex No. 215349

UNIX is a trademark of Bell Laboratories

Within The USA

IBC/ DISTRIBUTION

1140 36th Street Suite 212
Ogden, Utah 84403
(801) 621-2294

Circle No. 202 on Inquiry Card